McMaster University ENGINEERING

W BOOTH SCHOOL OF ENGINEERING PRACTICE AND TECHNOLOGY

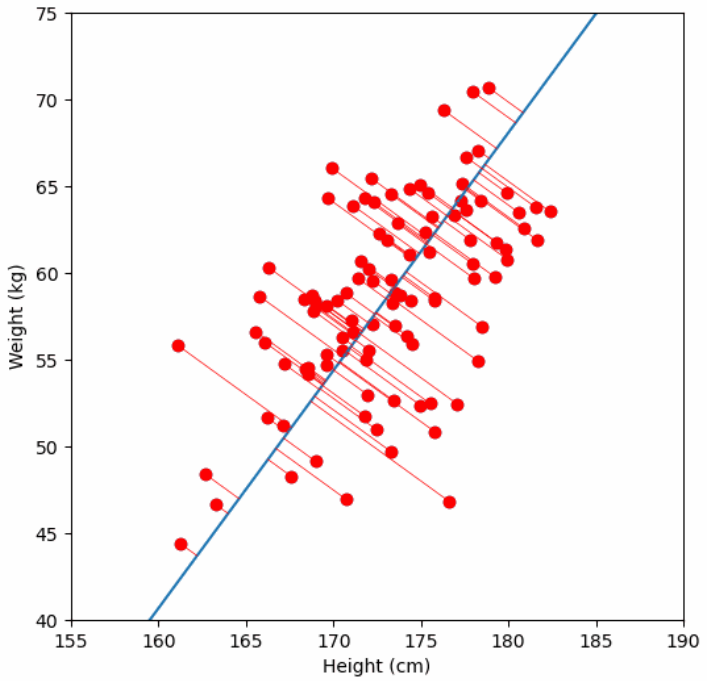
SEP 785: Machine Learning

Lecture 5: Linear Models for Regression

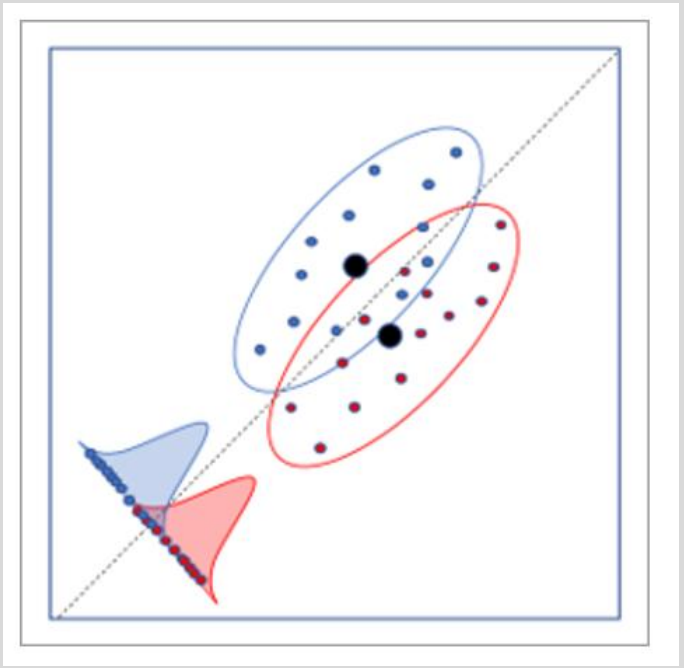
Instructor: Dr. Dalia Mahmoud, PhD
(Mechanical Engineering, McMaster University)
Email: mahmoudd@mcmaster.ca

Recap

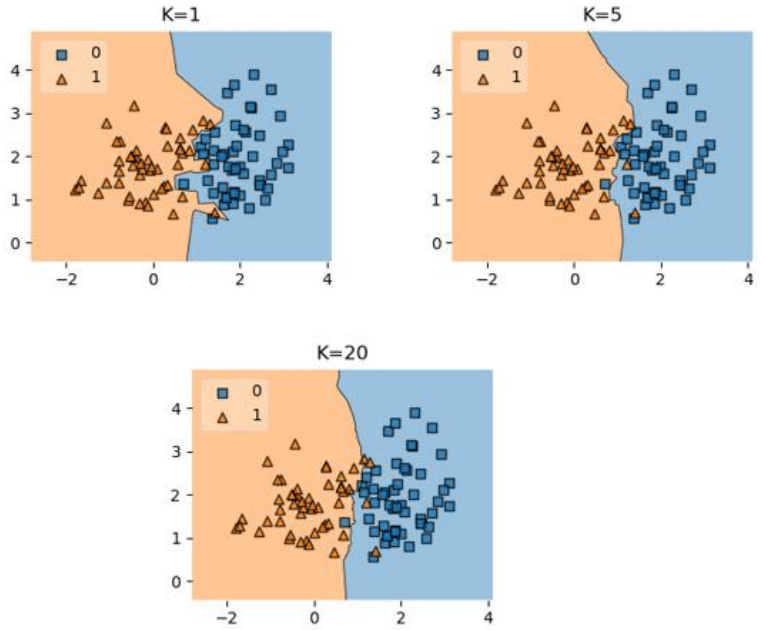
PCA



LDA



K_NN



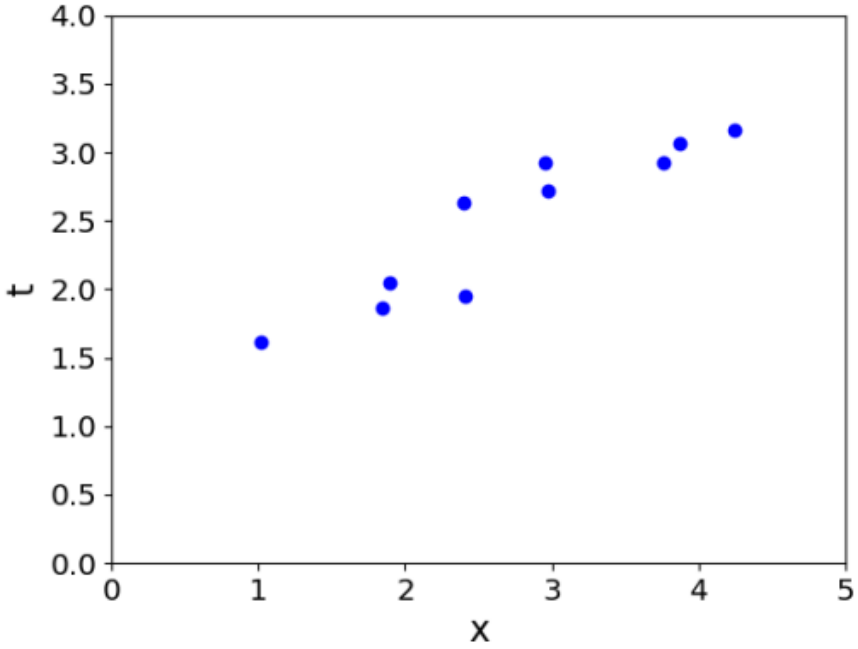
Intended Learning Outcomes

1. Explain the concept and cost function of linear regression.
2. Evaluate model performance using metrics like Mean Squared Error (MSE).
3. Understand and implement gradient descent to minimize the cost function.
4. Differentiate between Mini-Batch and Stochastic Gradient Descent (SGD).
5. Apply L1 (Lasso) and L2 (Ridge) and ElasticNet regularization to prevent overfitting.

Contents

Linear Models

- Input x ∈ X (vector of features or covariates)
- Target $t \in T$ (response, outcome, output, or class)
- The goal is to learn a function $f: X \rightarrow T$ such that the predicted output y=f(x) approximates the true target t



Linear Models

In linear regression, we use a **linear function** of the features:

$$x = (x_1, x_2, \ldots, x_D) \in \mathbb{R}^D$$

to make predictions y of the target value t ∈ R

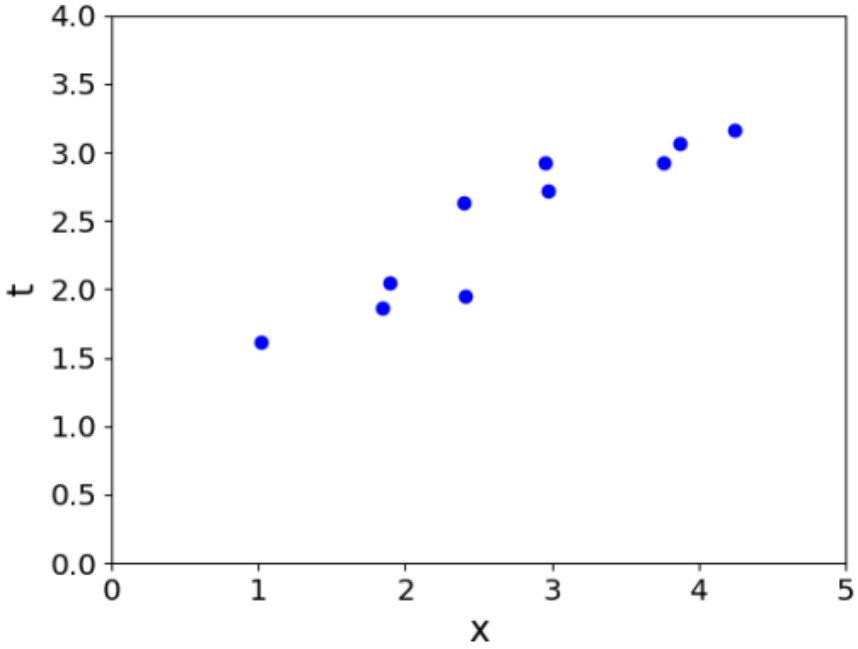
$$y = f(x) = \sum_{j=1}^{D} w_j x_j + b$$

y is the predicted output.

w is the vector of **weights**.

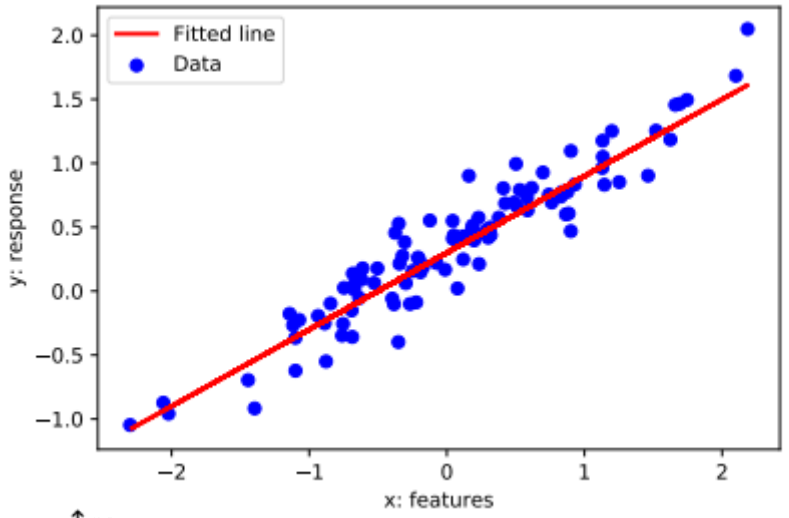
b is the **bias** (or intercept).

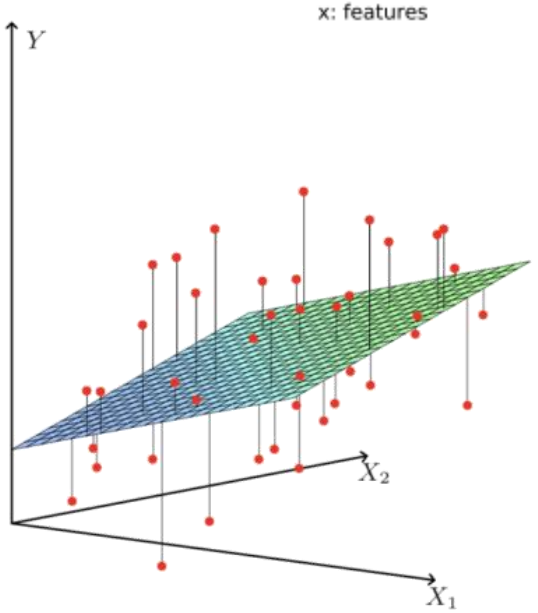
w and **b** together are the **parameters** of the model.



Linear Models

- If we have only 1 feature: y = wx + b where w, x, b ∈ R. y is linear in x.
- If we have D features: y = w>x + b where w, x ∈ R D, b ∈ R y is linear in x.
- Relation between the prediction y and inputs x is linear in both cases.

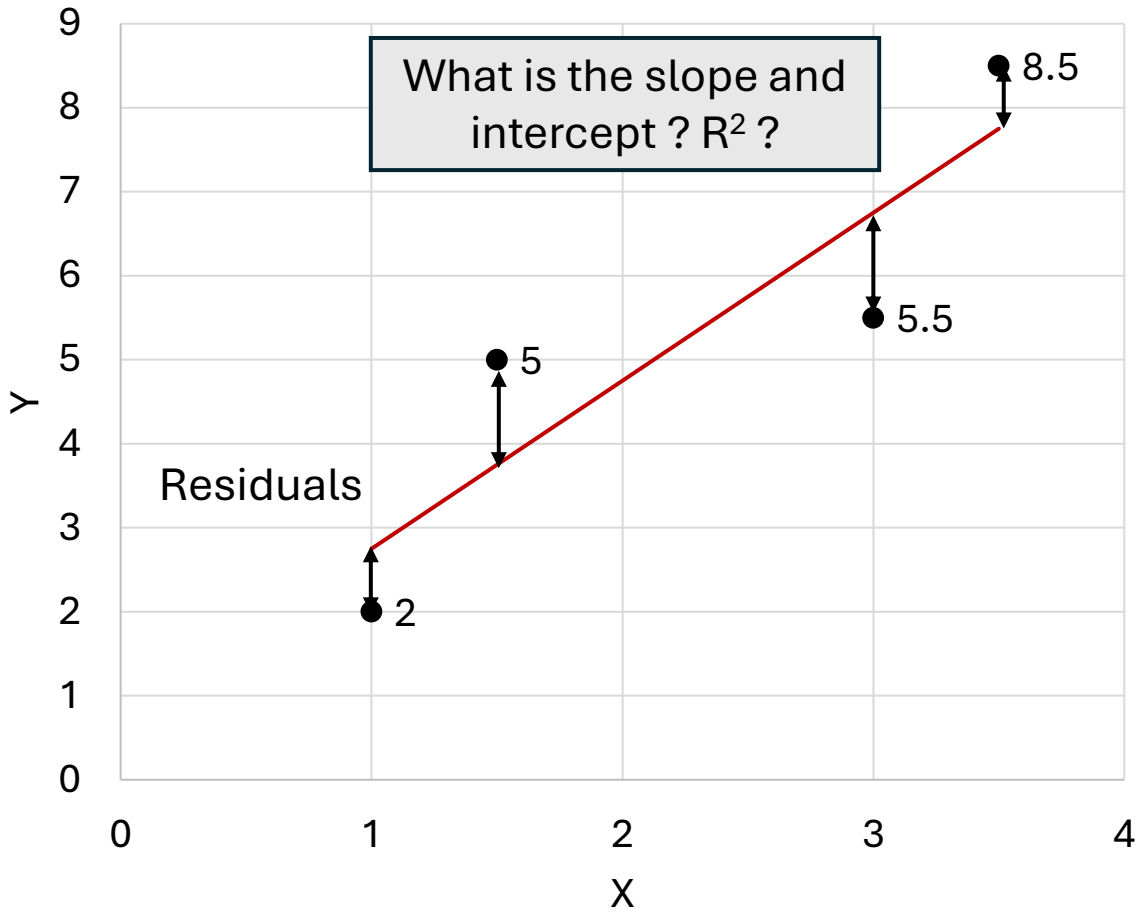




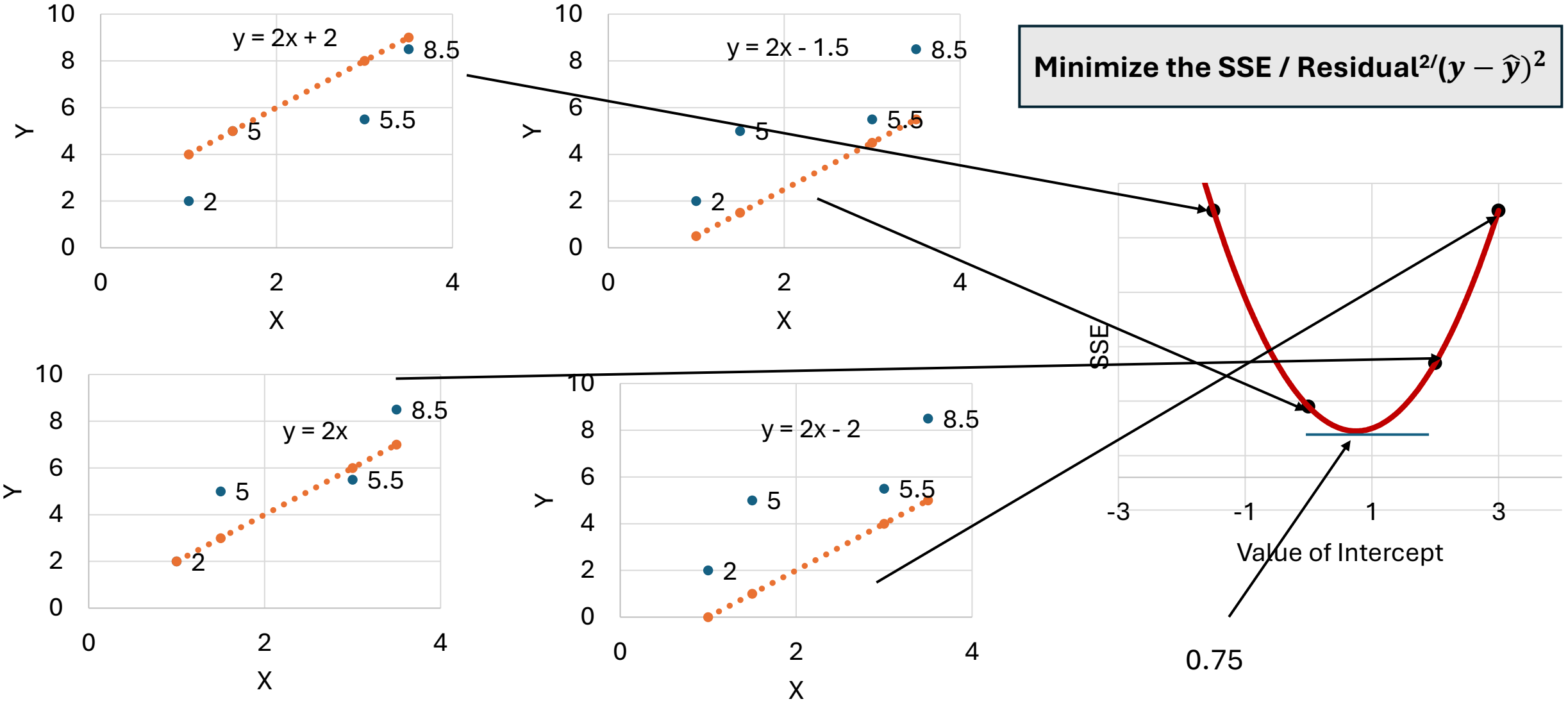
Method of Least Squares (OLS)

X	Y	$\hat{y}$	$(y-\hat{y})$	$(y-\hat{y})^2$
1	2	??	??	??
1.5	5			
3	5.5			
3.5	8.5			

$\hat{y}$ is the estimated value from equation
Residual= y- $\hat{y}$



Method of Least Squares (OLS)



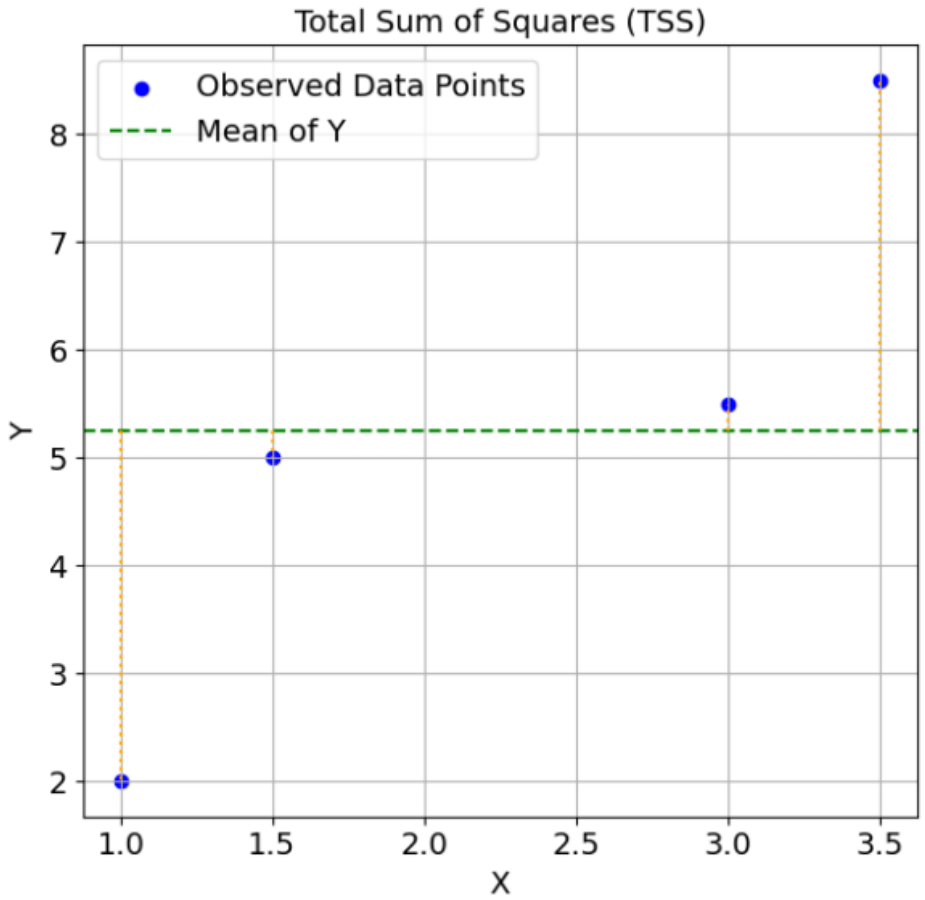
Total Sum of Squares (TSS)

- The sum of squares total represents the total variability in the dependent variable, including the portions of variability both explained and unexplained by the regression model.

$$TSS = \sum (Y_i - \bar{Y})^2$$

$$TSS = (2 - 5.25)^2 + (1.5 - 5.25)^2 + (3 - 5.25)^2 + (3.5 - 5.25)^2 = 21.25$$

X	Y
1	2
1.5	5
3	5.5
3.5	8.5
Average $\bar{y}$	5.25

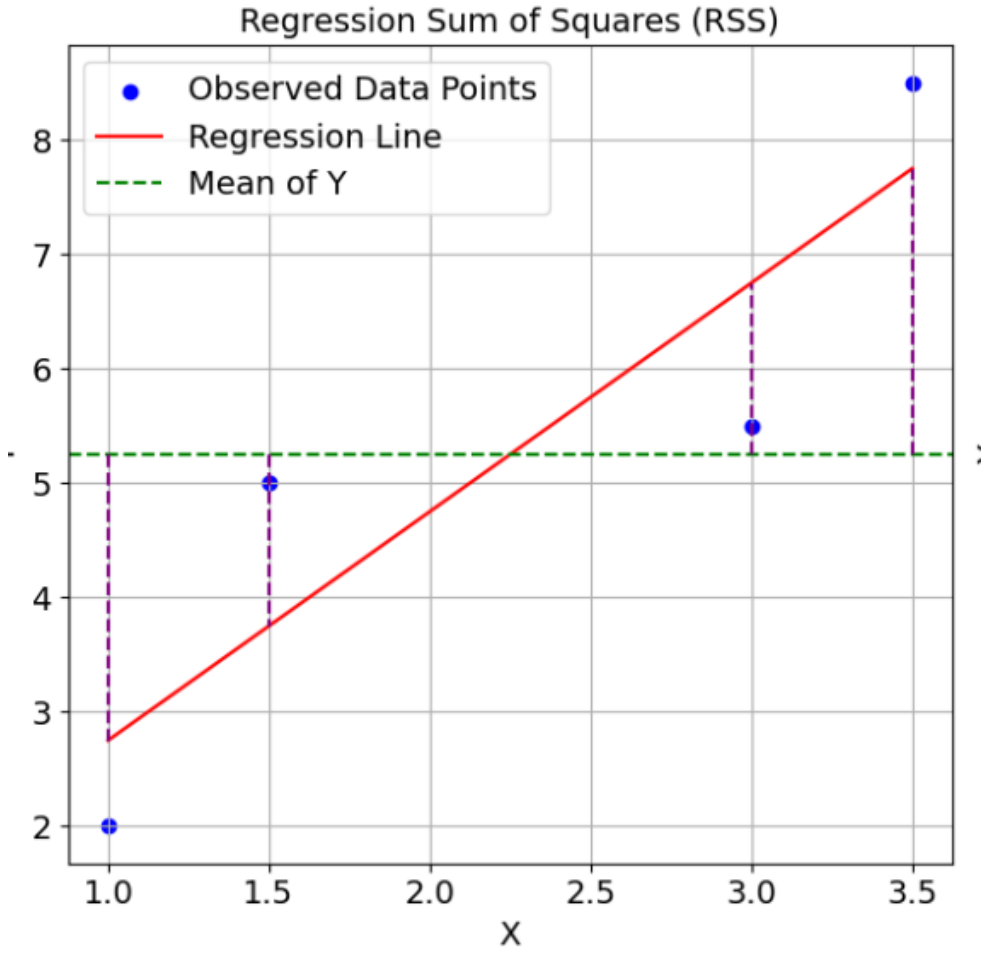


Regression Sum of Squares (RSS)

- The sum of squares regression captures the portion of the total variability in the dependent variable explained by the regression model.

X	Y	$\hat{y}$
1	2	2.75
1.5	5	3.75
3	5.5	6.75
3.5	8.5	7.75
Average $\bar{y}$	5.25	

$$RSS = \sum (\hat{Y}_i - \bar{Y})^2$$

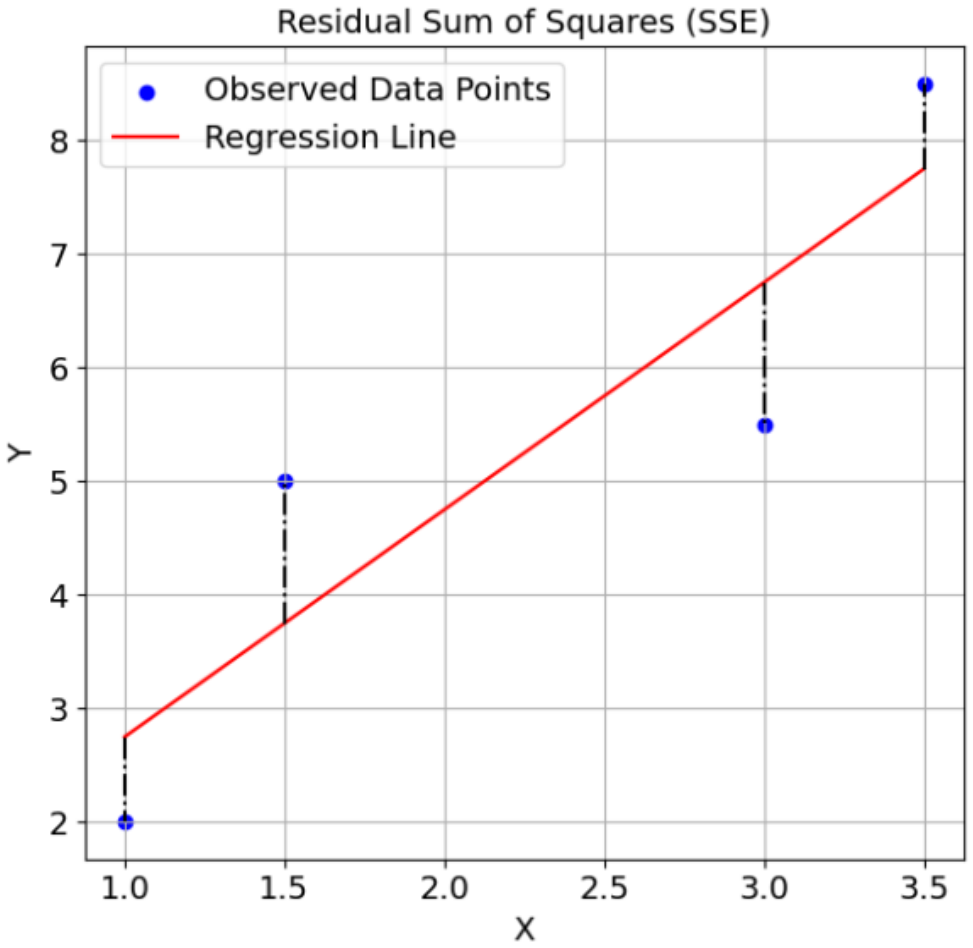


$$RSS = (2.75 - 5.25)^2 + (3.75 - 5.25)^2 + (6.75 - 5.25)^2 + (7.75 - 5.25)^2 = 17$$

Residual Sum of Squares (SSE)

- This measures how much of the total variation is **not explained** by the regression model.

X	Y	$\hat{y}$	$(y-\hat{y})$	$(y-\hat{y})^2$
1	2	2.75	-0.75	0.5625
1.5	5	3.75	1.25	1.5625
3	5.5	6.75	-1.25	1.5625
3.5	8.5	7.75	0.75	0.5625



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

$$SSE = 0.5625 + 1.5625 + 1.5625 + 0.5625 = 4.25$$

Summary

TSS is the total variability in the data.

RSS is the explained variability (from the model).

SSE is the unexplained variability (error).

TSS=RSS+SSE which means the total variability in the data is the sum of the explained and unexplained variability.

$$SST = (2 - 5.25)^2 + (1.5 - 5.25)^2 + (3 - 5.25)^2 + (3.5 - 5.25)^2 = 21.25$$

$$SSR = (5 - 5.25)^2 + (6 - 5.25)^2 + (9 - 5.25)^2 + (10 - 5.25)^2 = 17$$

$$SSE = 21.25 - 17 = 4.25$$

R^2 (Coefficient of Determination)

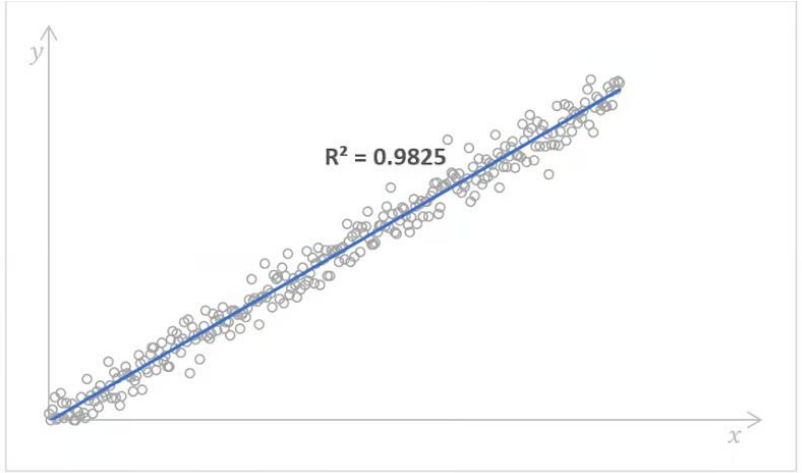
McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

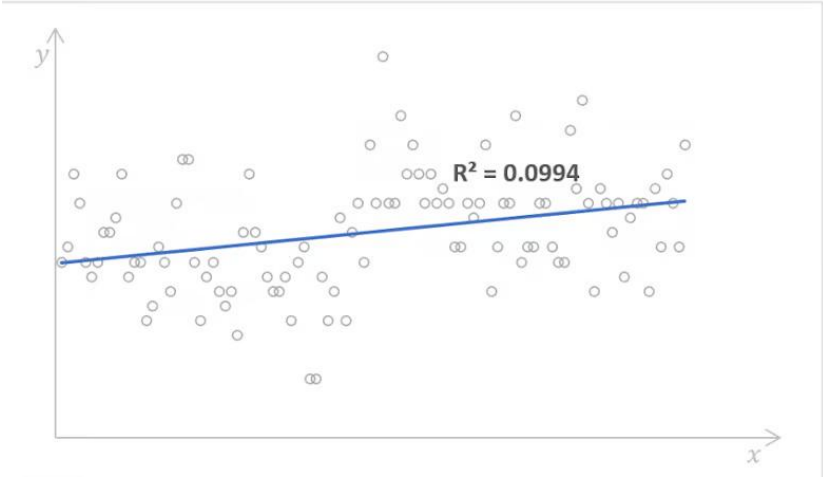
Is a statistical measure that represents the proportion of the variance in the dependent variable that is predictable from the independent variables.

It provides an indication of how well the regression model fits the data.

$$R^2 = \frac{SST - SSE}{SST}$$

$$R^2 = (21.25 - 4.25)/21.25 = 0.8$$





Minimizing the Sum of Squared Errors

- The goal is to find the values of www and w_0 and w that minimize the residual sum of squares (RSS), given by.

1. get the RSS equation
2. take the derivative of RSS with respect to w and set it equal to zero

$$y = w_0 + wx$$

$$RSS = \sum (y_i - (w_0 + wx_i))^2$$

$$\frac{\partial}{\partial w} RSS = \sum [-2x_i(y_i - w_0 - wx_i)] = 0$$

$$\sum x_i(y_i - w_0 - wx_i) = 0$$

$$\sum x_i y_i - w_0 \sum x_i - w \sum x_i^2 = 0$$

Recognize this ?

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

Minimizing the Sum of Squared Errors

3- Next, we differentiate RSS with respect to the intercept w_0 and w and set it equal to zero:

$$\frac{\partial}{\partial w_0} RSS = \sum [-2(y_i - w_0 - wx_i)] = 0$$

$$\sum (y_i - w_0 - wx_i) = 0$$

$$\sum y_i - nw_0 - w \sum x_i = 0$$

$$w_0 = \bar{y} - w\bar{x}$$

Method of Least Squares (OLS)

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

X	Y
1	2
1.5	5
3	5.5
3.5	8.5

$$\text{Slope} = \frac{\text{Cov}(x,y)}{\text{Var}(x)}$$

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

$$\text{Intercept} = \bar{y} - \text{slope} \cdot \bar{x}$$

$$\text{Slope} = \frac{(1-2.25)(2-5.25)+(1.5-2.25)(5-5.25)+(3-2.25)(5.5-5.25)+(3.5-2.25)(8.5-5.25)}{(1-2.25)^2+(1.5-2.25)^2+(3-2.25)^2+(3.5-2.25)^2} = \boxed{?}$$

$$\text{Intercept} = 5.25 - 2 \cdot 2.25 = 5.25 - 4.5 = \boxed{?}$$

Multiple Linear Regression Models

- In **multiple linear regression**, the relationship between the dependent variable y and the independent variables x_1, x_2 x_3,..., x_p is given by the equation:

$$y_i = w_0 + w_1 x_{i1} + w_2 x_{i2} + \cdots + w_p x_{ip} + \epsilon_i$$

- Where

y_i is the observed value of the dependent variable for the i-th observation,

$x_{i1}, x_{i2}, \ldots, x_{ip}$ are the values of the independent variables for the i-th observation,

w_0 is the intercept,

$w_1, w_2, \ldots, w_p$ are the coefficients (slopes) for each independent variable,

ϵ_i is the error term or residual for the i-th observation.

Multiple Linear Regression Models

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \qquad X = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix} \qquad w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_p \end{bmatrix}$$

Using a Matrix Formation the linear equation can now be written as

$$y = Xw + \epsilon$$

y is the vector of observed values,

X is the design matrix,

w is the vector of coefficients,

ϵ is the vector of residuals.

Multiple Linear Regression Models

- The residuals (errors) are the differences between the observed values and the predicted values. The predicted values y^ are given by

$$\hat{y} = Xw$$

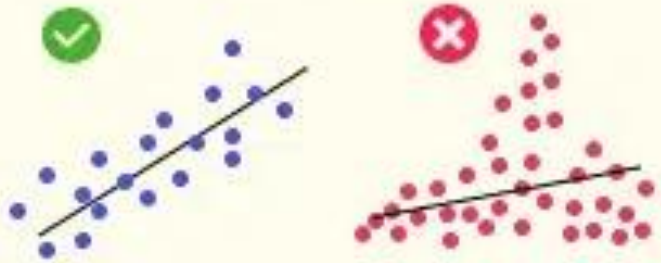
$$\epsilon = y - \hat{y} = y - Xw$$

$$RSS = \sum_{i=1}^{n} \epsilon_i^2$$

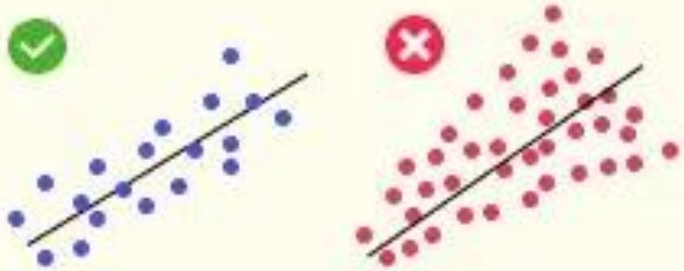
$$RSS = \epsilon^T \epsilon = (y - Xw)^T (y - Xw)$$

Assumptions of OLS

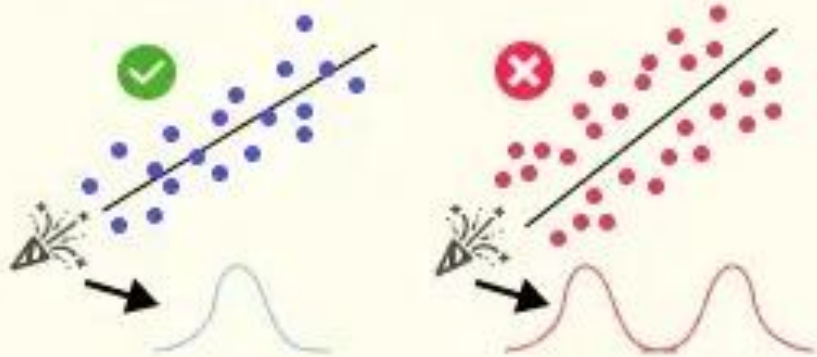
1. Linearity
(Linear relationship between Y and each X)



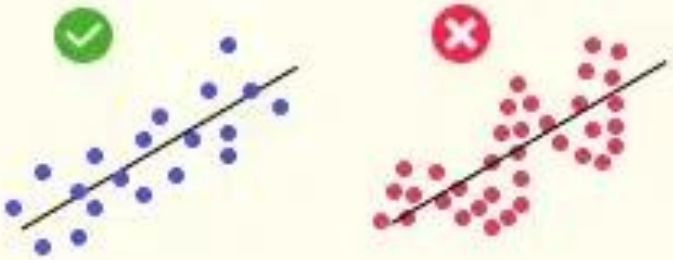
2. Homoscedasticity
(Equal variance)



3. Multivariate Normality
(Normality of error distribution)



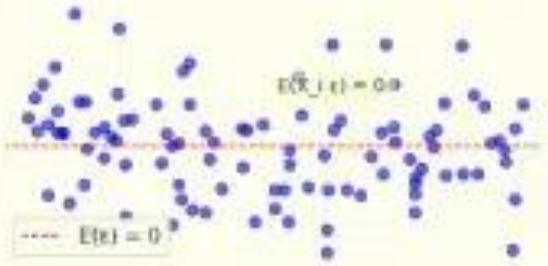
4. Independence
(of observations. Includes "no autocorrelation")



5. Lack of Multicollinearity
(Predictors are not correlated with each other)

$X_1 + X_2$ $X_1 \sim X_2$

6. Absence of endogeneity
(No correlation between predictors and errors.)

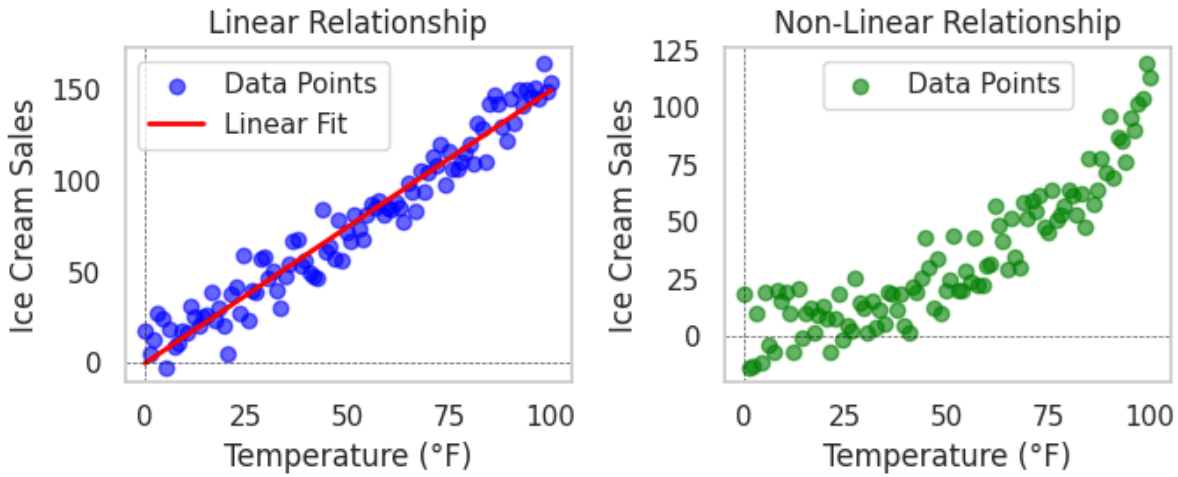


https://www.geeksforgeeks.org/assumptions-of-linear-regression/

Linearity

- Assumption: The relationship between the independent and dependent variables is linear.

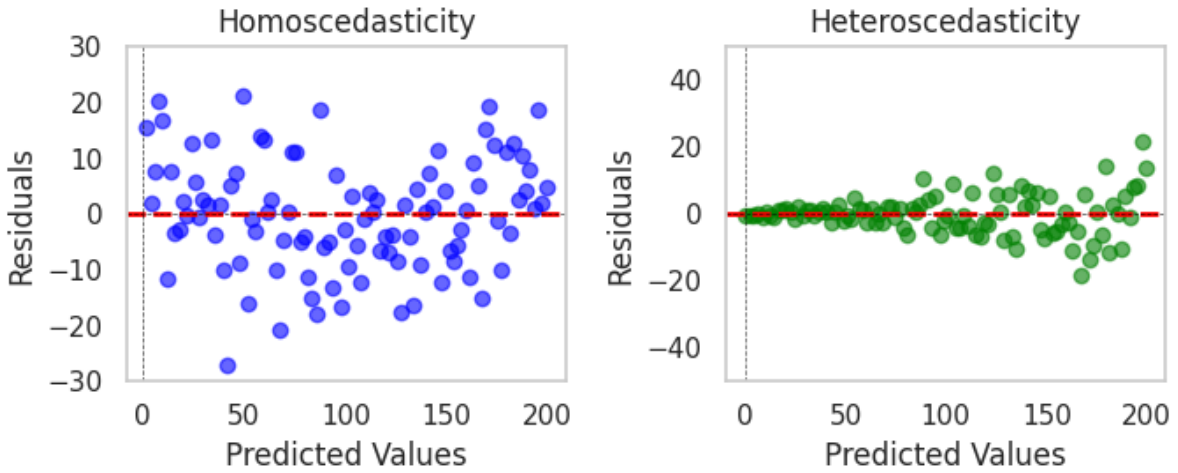
- This means that a change in the independent variable results in a proportional change in the dependent variable. This can be visually assessed using scatter plots or residual plots



Homoscedasticity of Residuals in Linear Regression

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

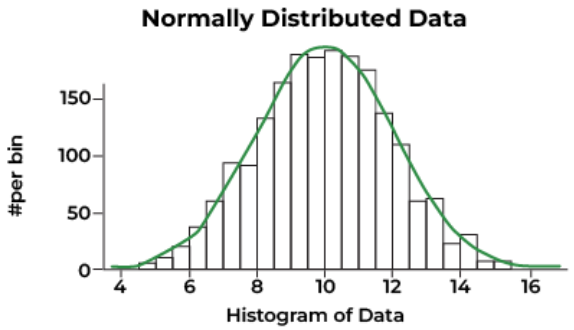
- Homoscedasticity means that the variance of the residuals (errors) is constant across all levels of the independent variable(s).
- If the assumption of homoscedasticity is violated (i.e., you have heteroscedasticity), it can lead to inefficient estimates of the regression coefficients and cause misleading inferences about the statistical significance of the predictors.

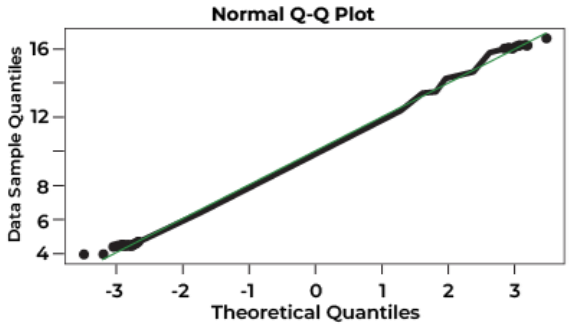


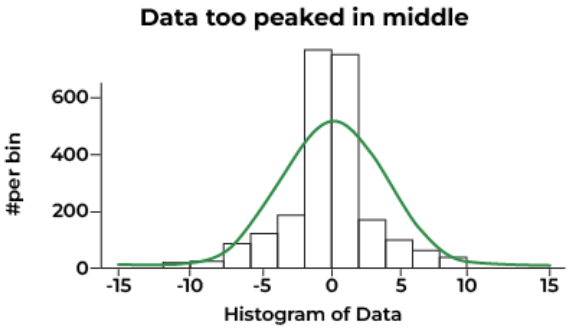
Normal Distribution

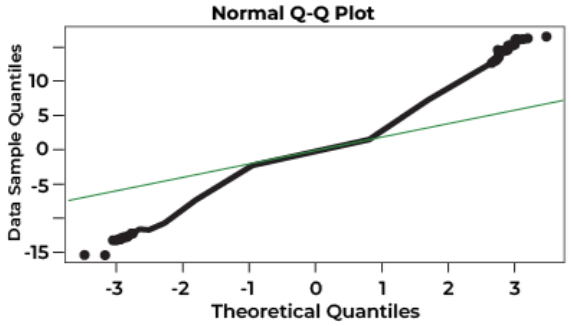
McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

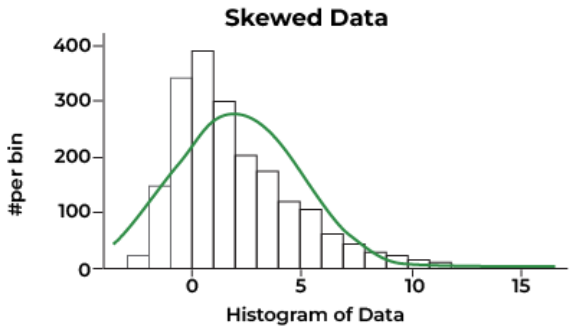
- It means that the residuals should follow a normal distribution when considering multiple predictors together.
- This assumption ensures that hypothesis tests, confidence intervals, and p-values are valid.

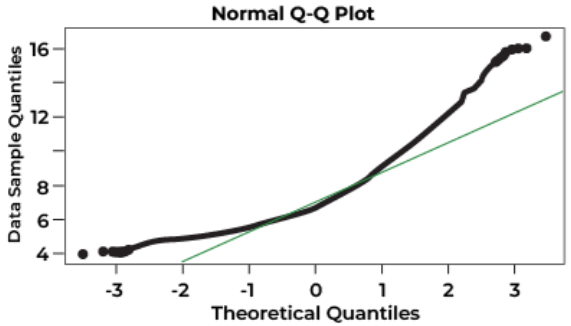






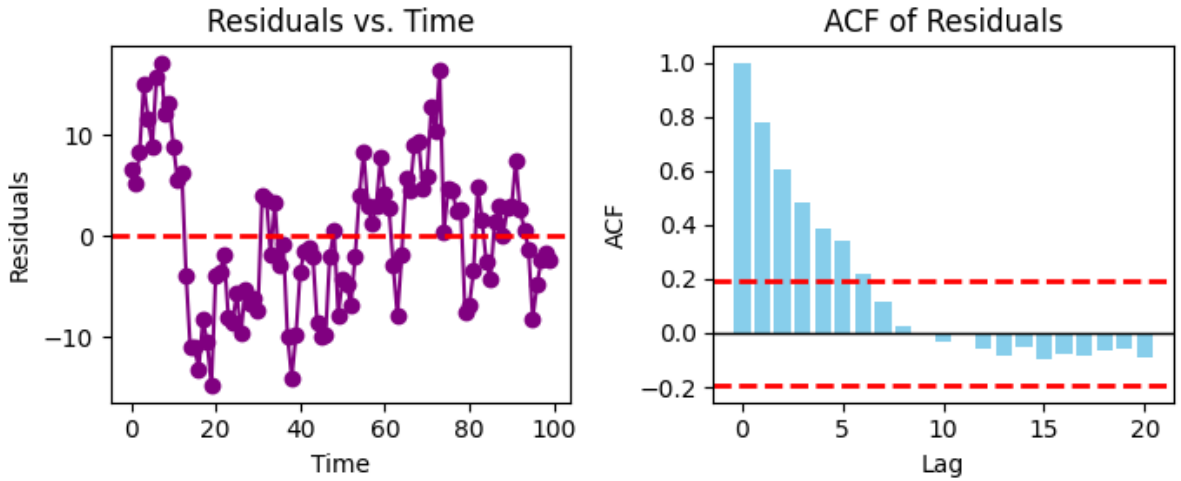






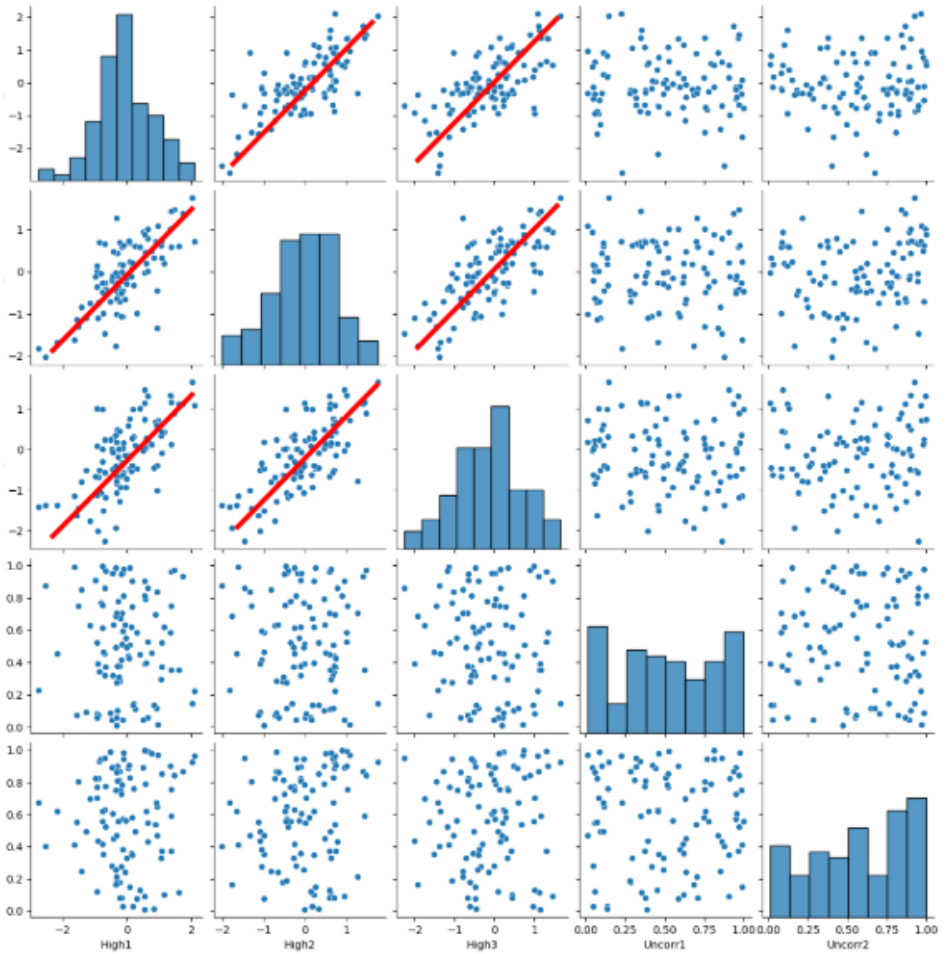
Independence of Errors

- It ensures that the residuals are not correlated with one another.
- This means that the error associated with one observation should not influence the error of any other observation.



Lack of Multicollinearity

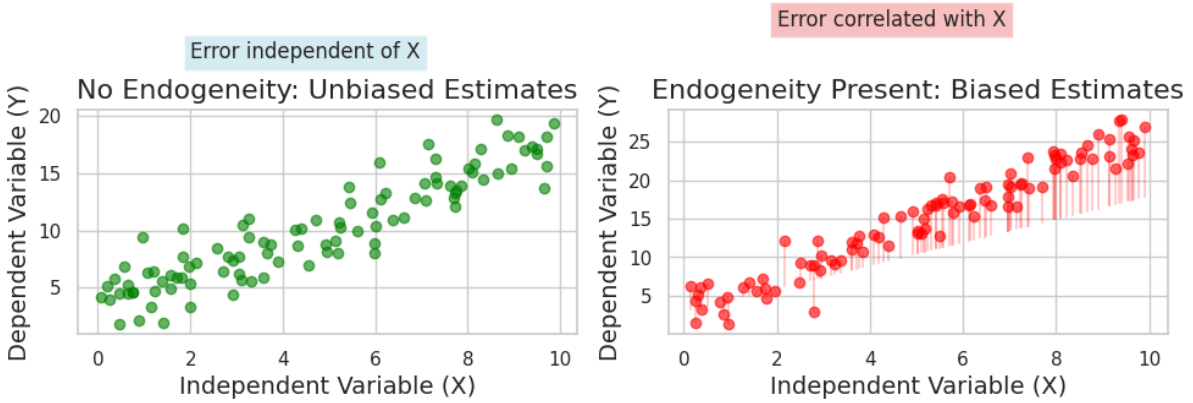
- Multicollinearity occurs when two or more independent variables in the model are highly correlated, leading to redundancy in the information they provide.
- This can inflate the standard errors of the coefficients, making it difficult to determine the effect of each independent variable



Absence of Endogeneity

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

- The assumption of no endogeneity states that the independent variables in the regression model should not be correlated with the error term.
- If this assumption is violated, it leads to biased and inconsistent estimates of the regression coefficients.



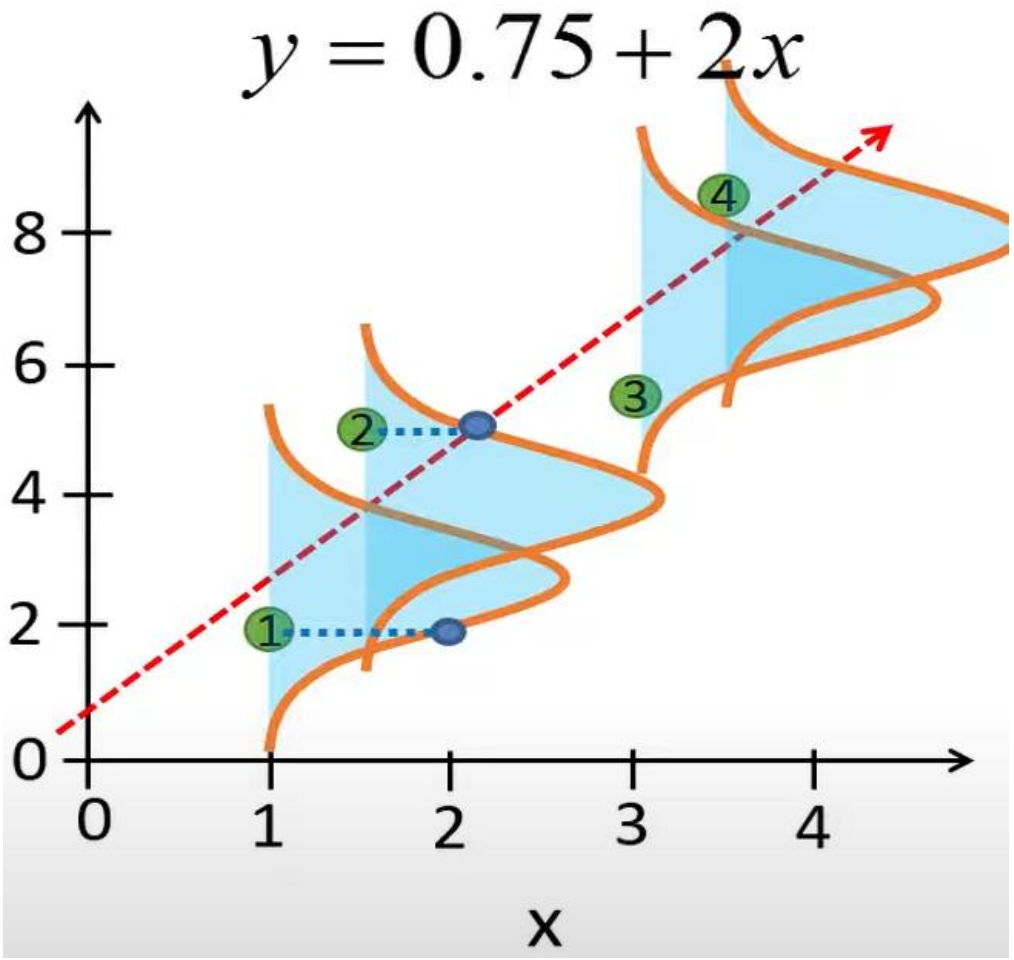
Contents

1. Linear regression
   - OLS
   - MLE
   - Gradient Decent
2. Regularization
   1. Ridge Regression
   2. Lasso Regression
   3. Elastic Net Regression
3. Logistic Regression
4. Support Vector Machines

Maximum Likelihood Estimation (MLE)

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

- Is a statistical method for estimating the parameters of a model that maximizes the likelihood of observing the given data.
- It finds the parameter values that make the observed data most probable under the assumed model.
- The main goal is to maximize the likelihood of observing the given data based on the model parameters.

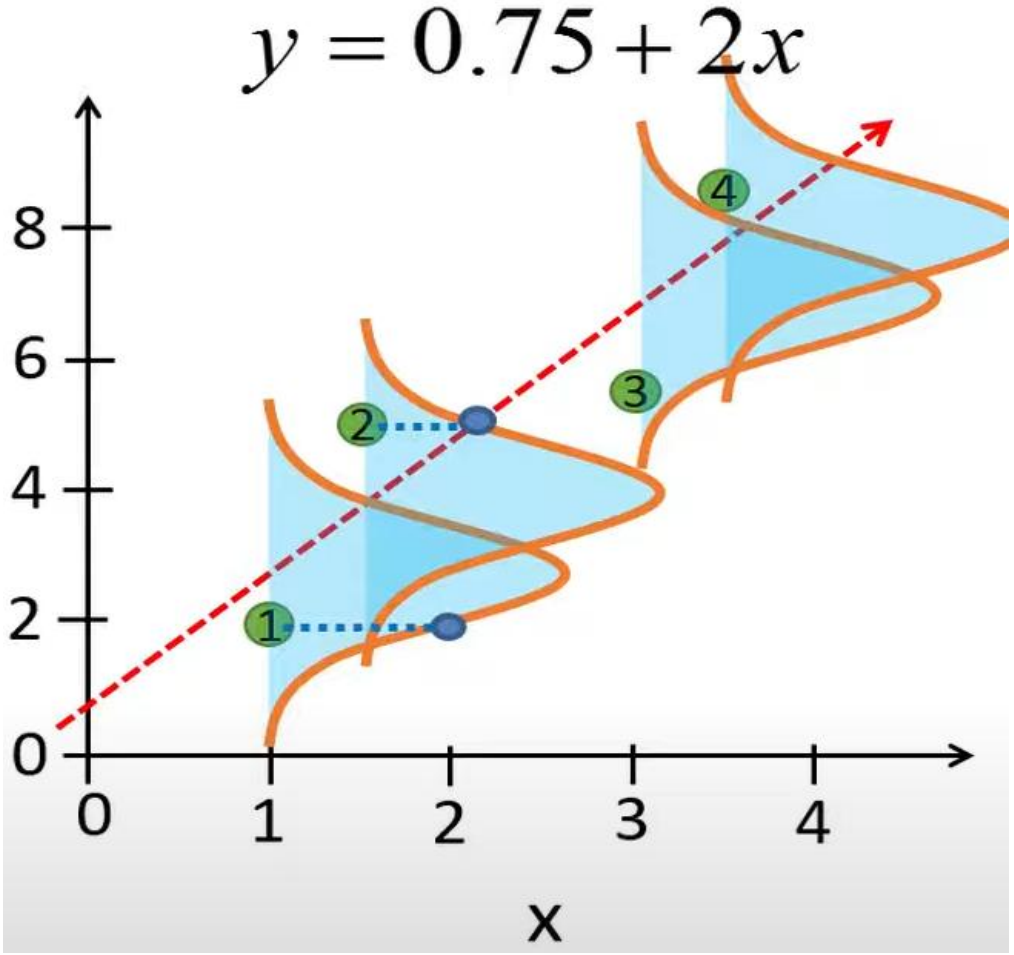


Maximum Likelihood Estimation (MLE)

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

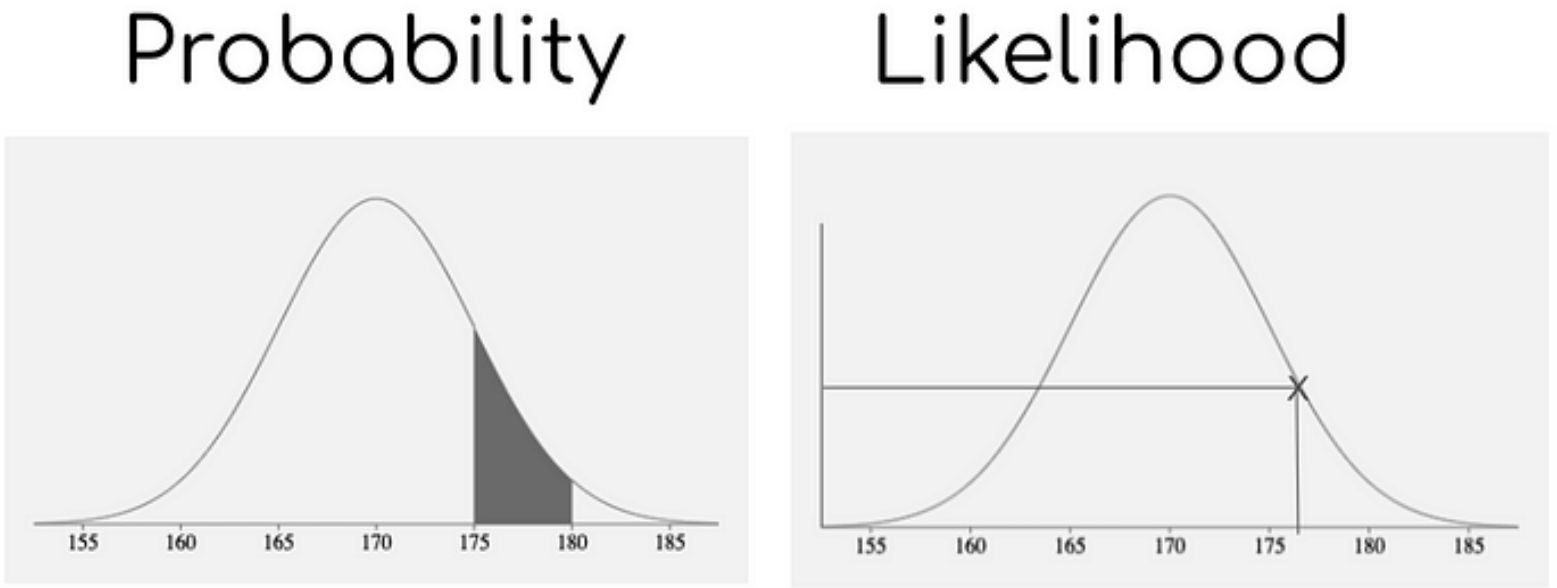
How it Works:

- Defines a likelihood function based on the data and model.
- Maximizes this function (or the log-likelihood) to find the best-fitting parameters.
- Likelihood is the probability of the observed data under a hypothetical scenario



Probability Vs Likelyhood

- https://youtu.be/pYxNSUDSFH4



Common Likelihood Functions

- **Normal**: Likelihood involves the product of Gaussian PDFs.
- **Binomial**: Likelihood involves the product of binomial PMFs.
- **Poisson**: Likelihood involves the product of Poisson PMFs.
- **Exponential**: Likelihood involves the product of exponential PDFs.
- **Uniform**: Likelihood involves the product of constant values over the data range.
- **Gamma**: Likelihood involves the product of gamma PDFs.

$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

$$f(x|n,p) = \binom{n}{x} p^x (1-p)^{n-x}$$

$$f(x|\lambda) = \frac{\lambda^x e^{-\lambda}}{x!}, \quad x = 0, 1, 2, \ldots$$

$$f(x|\lambda) = \lambda e^{-\lambda x}, \quad x \geq 0$$

$$f(x|a,b) = \frac{1}{b-a}, \quad a \leq x \leq b$$

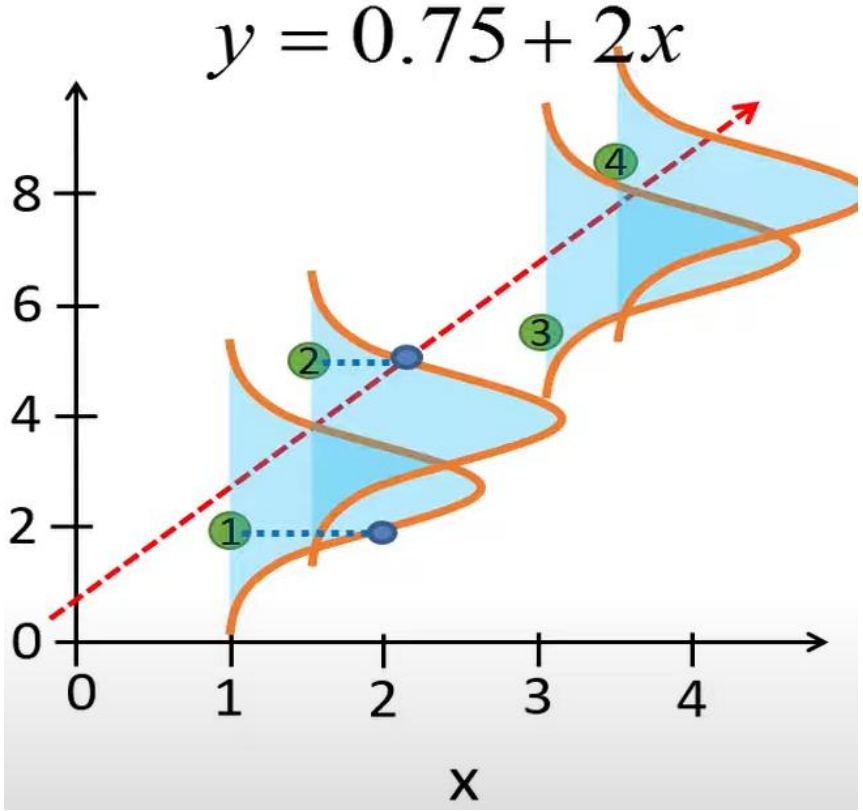
$$f(x|k,\lambda) = \frac{\lambda^k x^{k-1} e^{-\lambda x}}{\Gamma(k)}, \quad x \geq 0$$

Common Likelihood Functions

Data Type	Distribution	Example Application
Continuous (Unbounded)	Normal (Gaussian)	Heights, weights, errors in regression
Binary (0/1)	Bernoulli	Yes/No, Spam/Not Spam
Count Data	Poisson	Number of calls per day
Multiple Successes in Trials	Binomial	Number of defective items in a batch
Time Until Event	Exponential	Machine failure time
Generalized Time Until Event	Gamma	Reliability analysis
Classification	Logistic Regression	Predicting disease presence

Linear Regression (Method of Likelihood)

X	Y	Y-μ estimated	Residual	Residual²	L
1	2	2.75	-0.75	0.5625	0.301
1.5	5	3.75	1.25	1.5625	0.183
3	5.5	6.75	-1.25	1.5625	0.183
3.5	8.5	7.75	0.75	0.5625	0.301



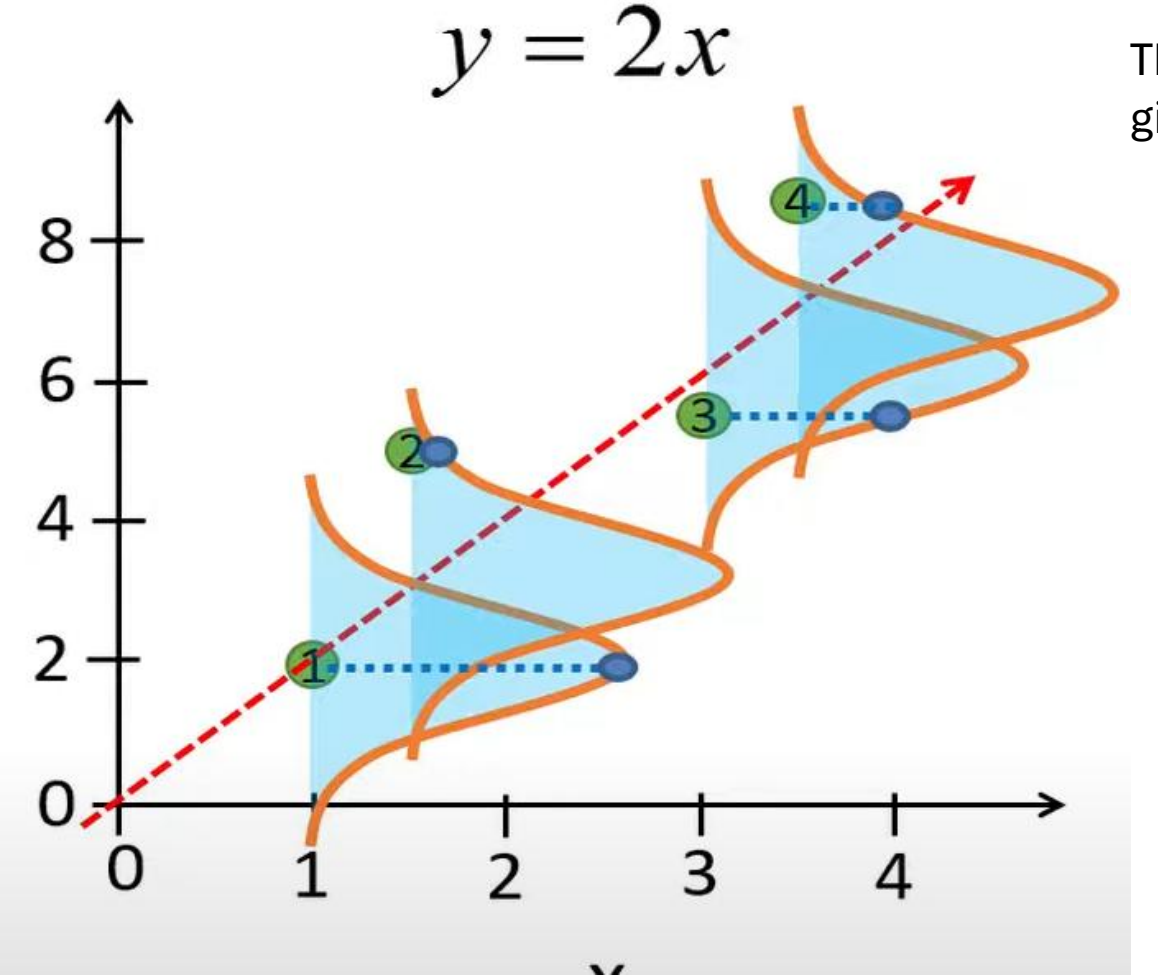
σ=1

$$f(Y)=\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(Y-\mu)^2}{2\sigma^2}}$$

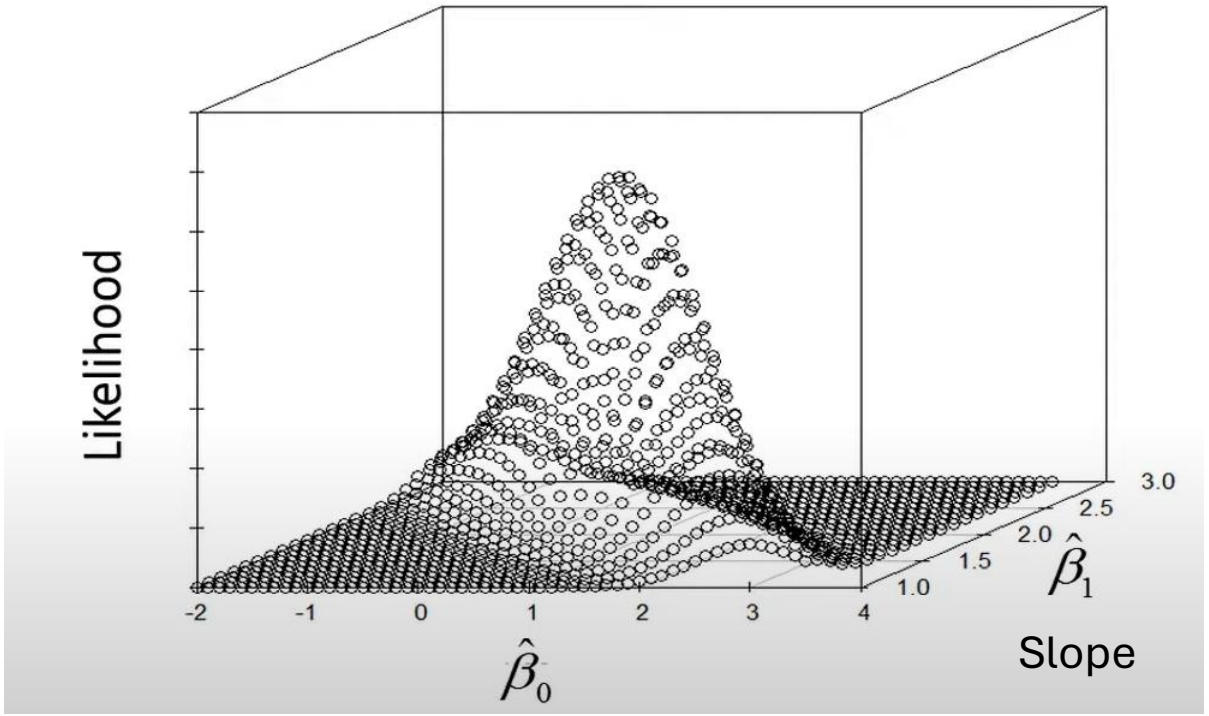
$$L(\hat{\beta}_0=0.75,\hat{\beta}_1=2,\hat{\sigma}=1)=\frac{1}{\sqrt{2\pi\cdot1^2}}e^{-\frac{(2-2.75)^2}{2\cdot1^2}}\cdot\frac{1}{\sqrt{2\pi\cdot1^2}}e^{-\frac{(5-3.75)^2}{2\cdot1^2}}\cdot\frac{1}{\sqrt{2\pi\cdot1^2}}e^{-\frac{(5.5-6.75)^2}{2\cdot1^2}}\cdot\frac{1}{\sqrt{2\pi\cdot1^2}}e^{-\frac{(8.5-7.75)^2}{2\cdot1^2}}=0.301\cdot0.183\cdot0.183\cdot0.301=0.003$$

Linear Regression (Method of Likelihood)

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

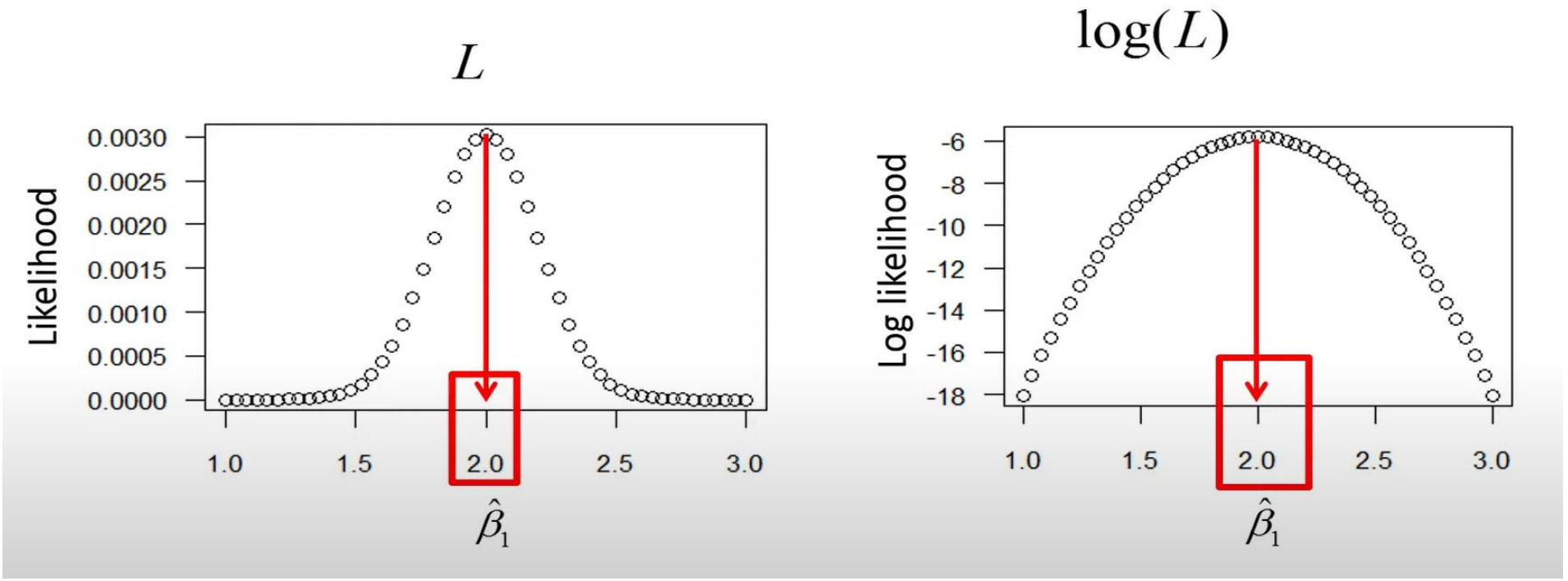


The main goal is to maximize the likelihood of observing the given data based on the model parameters.



Log-likelihood

- Taking the log of the likelihood function simplifies calculations, particularly when dealing with large datasets or complex models, and makes optimization (like MLE) more tractable and numerically stable.



When to use MLE

- **Probabilistic Modeling:** When you need a general **probability-based** framework for estimation.
- **Non-Normal Errors:** If errors are not normally distributed (e.g., logistic regression assumes a Bernoulli distribution for binary outcomes).
- **More Flexible Distributions:** MLE allows for modeling using different distributions (e.g., Poisson, Exponential, Gamma) depending on the data.
- **Large Sample Efficiency:** MLE is asymptotically efficient, meaning it performs well in large sample sizes.
- **Likelihood-Based Inference:** MLE naturally provides measures like confidence intervals, likelihood ratio tests, and Bayesian priors.

Aspect	MLE	OLE
Objective	Maximize likelihood of data given model params	Minimize sum of squared residuals
Assumptions	Known probability distribution (e.g., normal)	Linear relationship, constant variance
Estimation Method	Optimizes likelihood/log-likelihood	Solves linear equations directly
Efficiency	Flexible for complex, non-linear models	Efficient for linear models with normal errors
Robustness	Robust for non-linear and non-normal models	Sensitive to outliers and normality assumptions
Bias	Can be biased or unbiased based on sample size	Unbiased if assumptions hold

Contents

Loss Functions

The **loss function** is a mathematical function that quantifies the error in predictions.

Mean Squared Error (MSE):

- Used in OLS regression.
- Penalizes large errors more due to squaring.

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

- Cross-Entropy Loss (Log Loss):

$$\mathcal{L} = -\sum_{i=1}^{n}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right]$$

Mean Absolute Error (MAE):

- Less sensitive to outliers than MSE.

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

y_i = Actual observed value of the ith data point (ground truth).

$\hat{y}_i$ = Predicted value of the ith data point from the model.

Loss Functions

- The loss depends on the values of w_0 and w.
- To minimize the loss, you need to **adjust** w_0 and w to find the best-fitting line for the data.

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

$$\hat{y}_i = w_0 + wx_i$$

$$\text{MSE}(w_0, w) = \frac{1}{n}\sum_{i=1}^{n}(y_i - (w_0 + wx_i))^2$$

w_0 is the **intercept** (or bias term),

w is the **slope** (or weight),

x_i is the input feature for the ith data point.

How to solve Loss Functions

- **Analytical solution** (for simple linear models) → Solve for parameters using **derivatives**.
- **Gradient Descent** (for complex models) → Update model parameters iteratively using the gradient.

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

$$\hat{y}_i = w_0 + wx_i$$

$$\text{MSE}(w_0, w) = \frac{1}{n}\sum_{i=1}^{n}(y_i - (w_0 + wx_i))^2$$

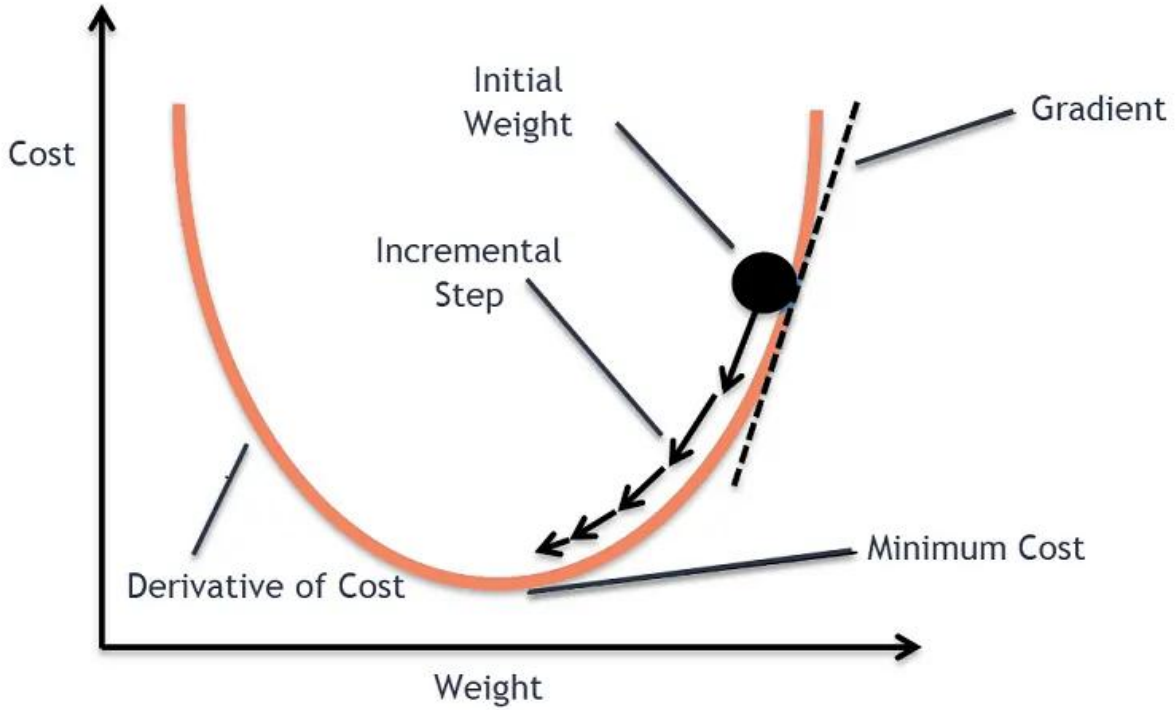
w_0 is the **intercept** (or bias term),

w is the **slope** (or weight),

x_i is the input feature for the ith data point.

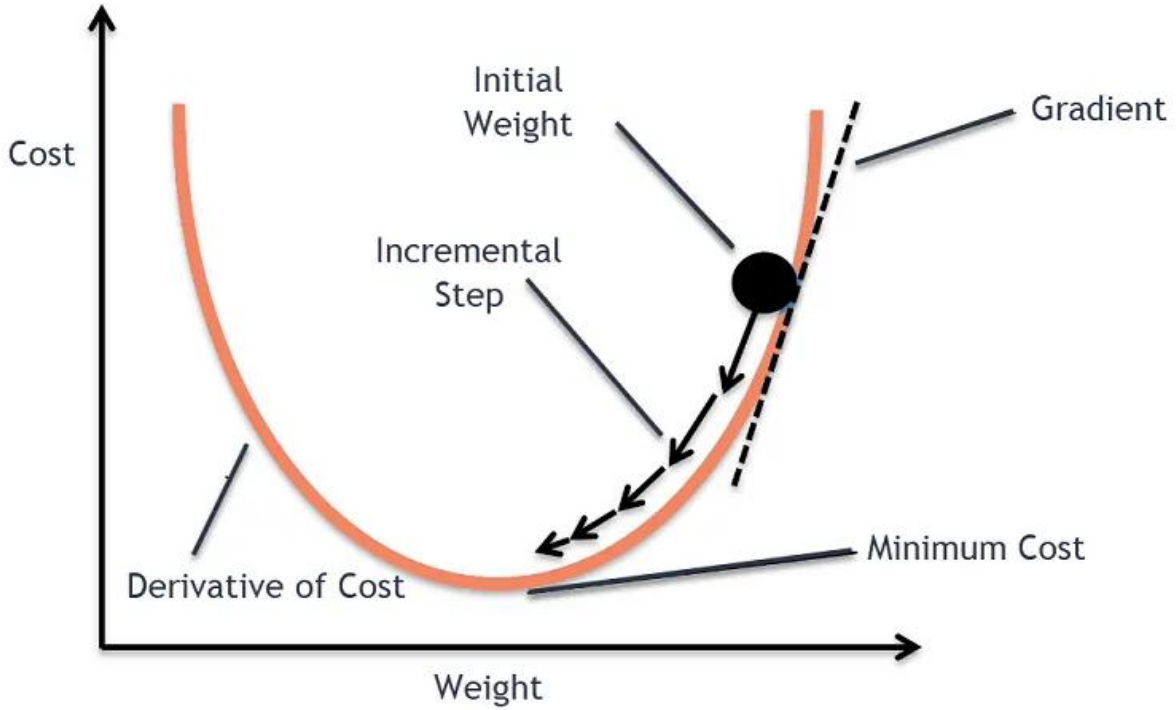
What is Gradient Descent

- Is an optimization algorithm used to minimize a **loss function** (or cost function) by iteratively adjusting the model's parameters (like weights w_0 and w) in the direction that reduces the loss.



How to use it Gradient Descent

1. Choose the Cost/Loss Function.
2. Initialize Parameters: Randomly initialize the model parameters
3. Set Learning Rate (α): The learning rate controls the size of the steps taken towards the minimum.
   - If it's too small, the algorithm might converge slowly; if it's too large, the algorithm might overshoot the minimum.
   - A common starting point is to set α between 0.01 and 0.1.

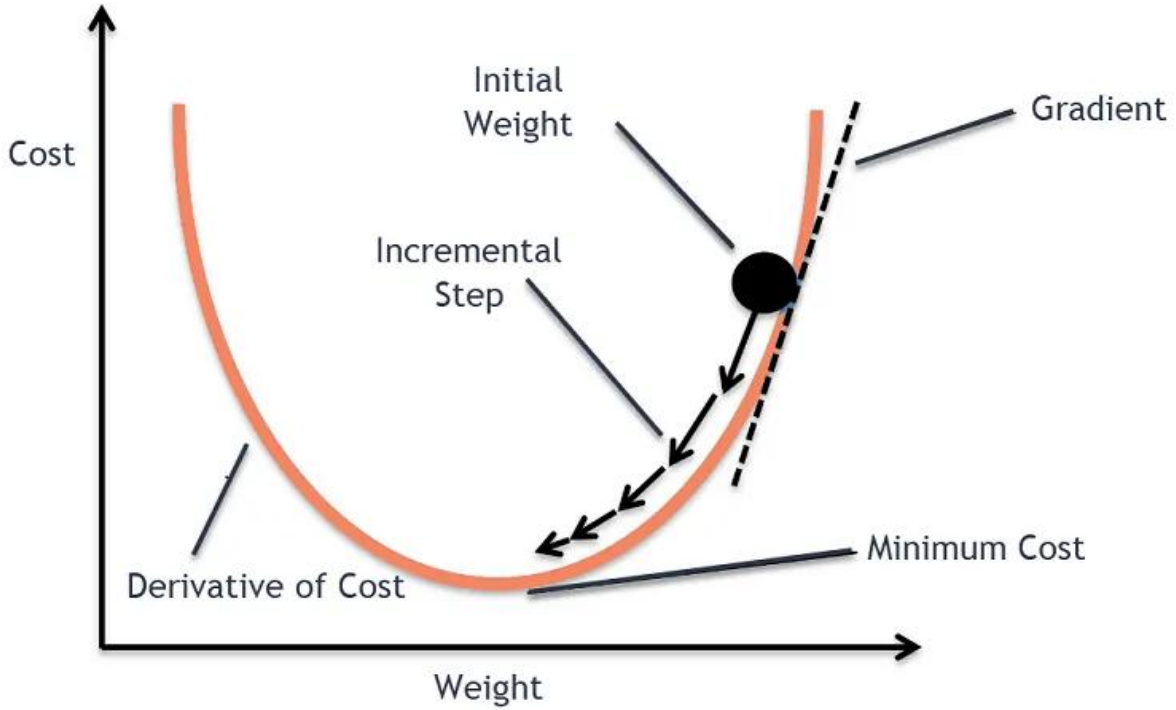


How to use it Gradient Descent

4. Compute the Gradient: The gradient is the partial derivative of the cost function with respect to each parameter

   The gradient represents the direction and rate of change of the cost function in each parameter dimension.

5. Update the Parameters: Once the gradient is computed, update the parameters in the direction opposite to the gradient (to minimize the cost).
6. Repeat Until Convergence: Repeat steps 4 and 5 for a set number of iterations or until the change in the cost function between iterations is smaller than a predefined threshold



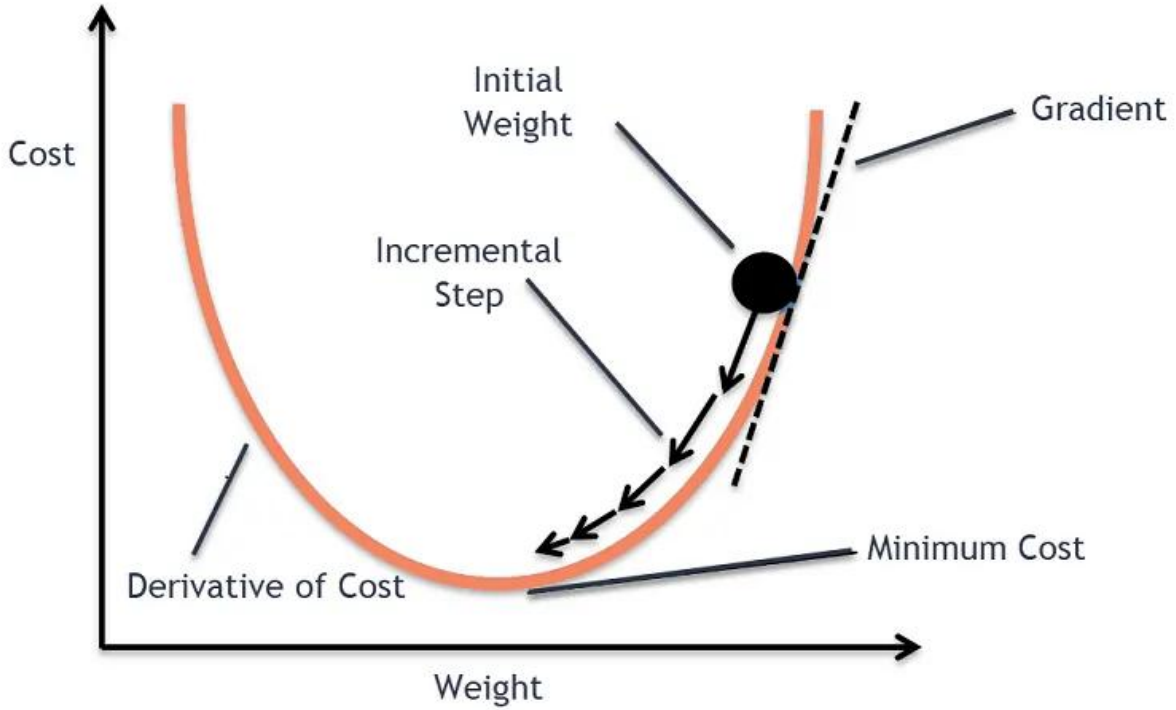
How to use it Gradient Descent

7-Stopping Criteria:

Fixed Number of Iterations: Set a predefined number of iterations, e.g., 1000 or 5000.

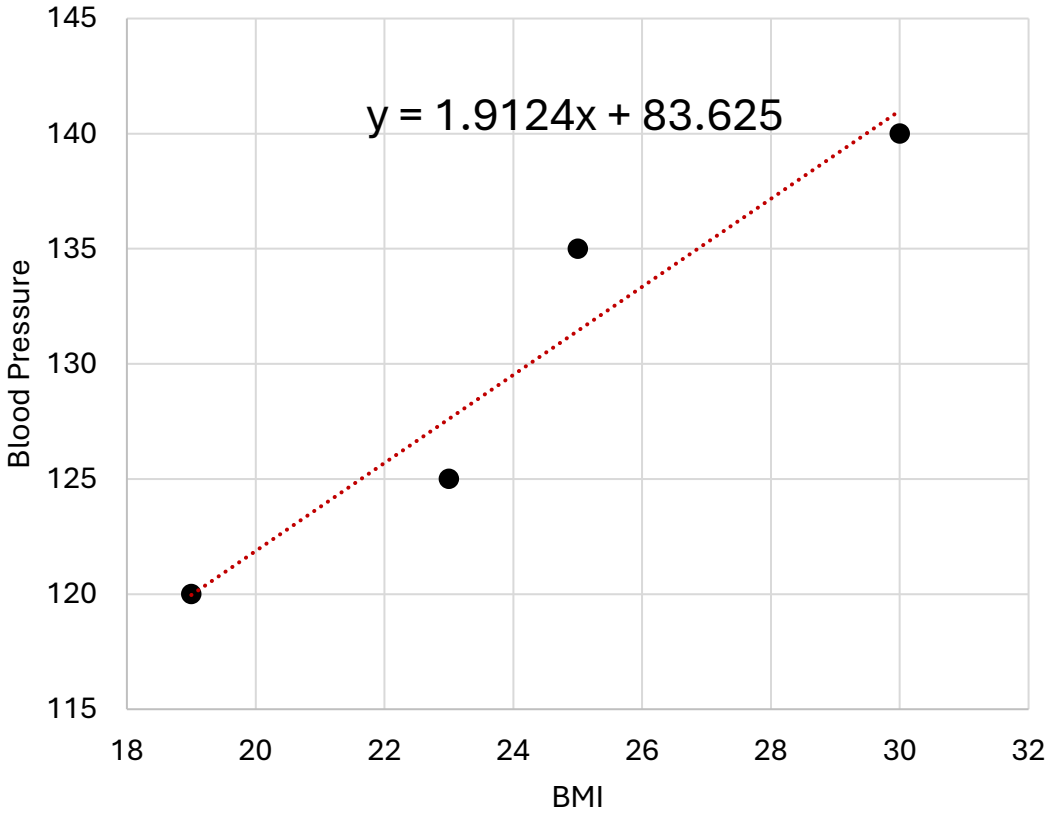
Convergence Criterion: Stop when the change in the cost function or parameter values is smaller than a threshold.

Learning Rate Decay: Sometimes the learning rate is gradually reduced to improve convergence as the model approaches the optimal solution.

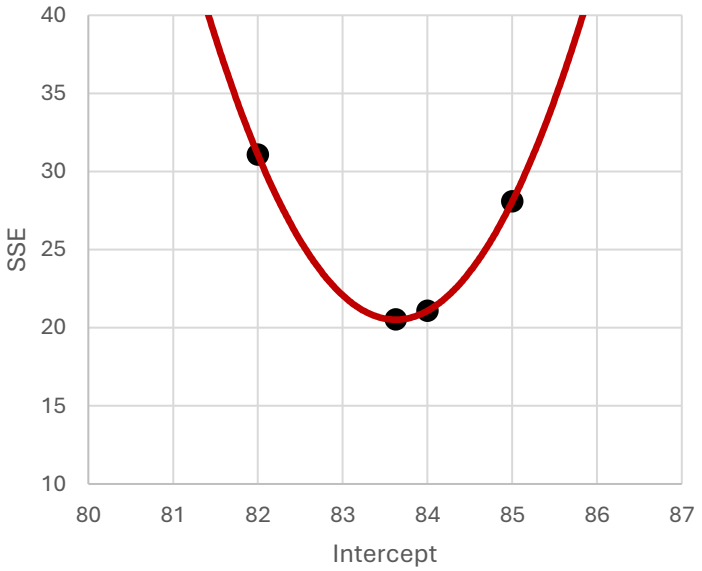


Gradient Descent Example

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology



BMI	Blood Pressure	$\hat{y}$	$(y - \hat{y})^2$
19	120		
25	135		
23	125		
30	140		



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

Gradient Descent Example

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

1. Choose the Cost/Loss Function.

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

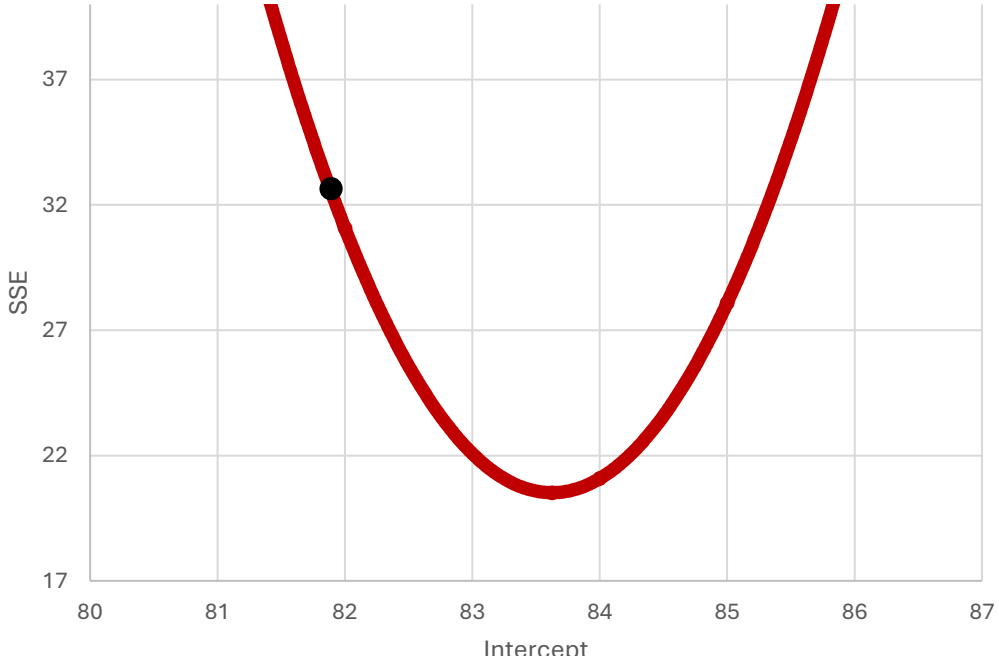
$$y = b_0 + b_1 x$$

2-Initialize Parameters:

b_o= 82
b_1= 1.912

3-Set Learning Rate (Y):

$$\gamma = 0.001$$



Gradient Descent Example

4. Compute the Gradient:

$$\text{SSE} = \sum_i^n (y_i - \hat{y}_i)^2$$

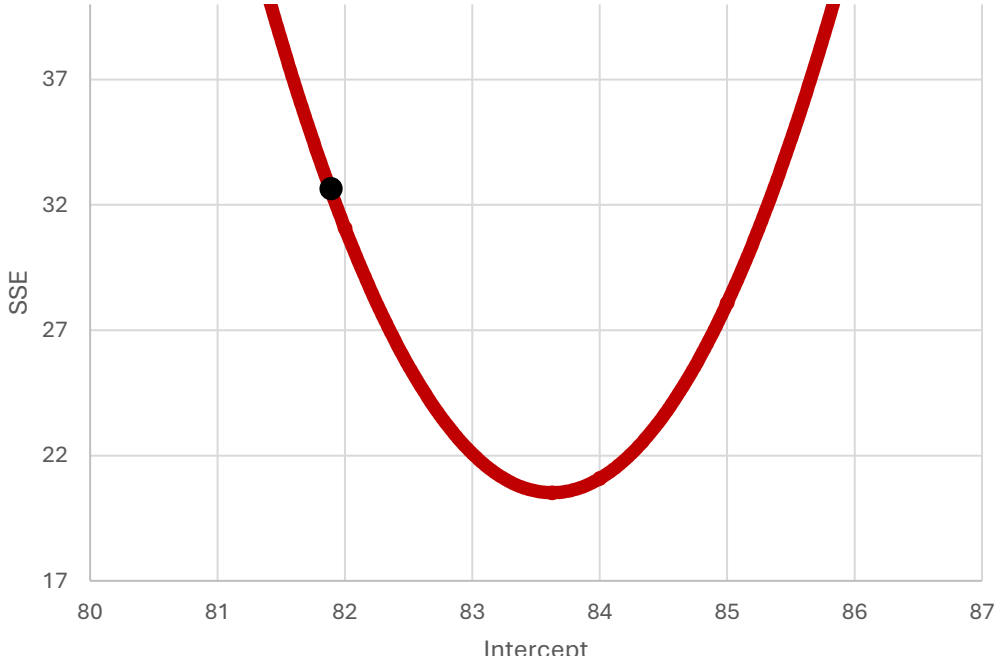
$\gamma = 0.001$

$y = b_0 + b_1 x$

b_o= 82
b_1= 1.912

$$\text{SSE} = \sum_i^n (y_i - (b_0 + b_1 x_i))^2$$

$$\frac{\partial \text{SSE}}{\partial b_0} = -2 \sum_i^n (y_i - (b_0 + b_1 x_i))$$



Gradient Descent Example

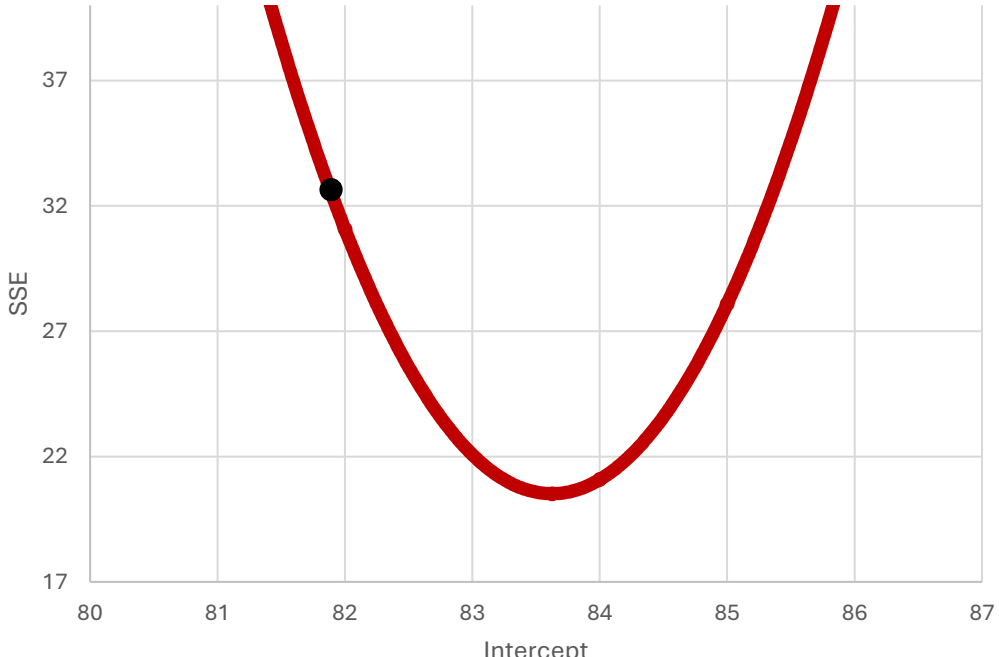
5. Update the parameters

$$\text{SSE} = \sum_i^n (y_i - \hat{y}_i)^2$$

$$y = b_0 + b_1 x$$

$\gamma = 0.001$

b_o= 82
b_1= 1.912



$$\text{SSE} = \sum_i^n (y_i - (b_0 + b_1 x_i))^2$$

$$\frac{\partial \text{SSE}}{\partial b_0} = -2 \sum_i^n (y_i - (b_0 + b_1 x_i))$$

$$b0_{new} = b0_{old} - \gamma \nabla f(b0_{old})$$

$$b0_{new} = 82 - 0.001 \nabla f(82)$$

$$\frac{\partial \text{SSE}}{\partial b_0} = -2((120 - 82 - 1.912 \cdot 19) + (135 - 82 - 1.912 \cdot 25)) + (125 - 82 - 1.912 \cdot 23)) + (140 - 82 - 1.912 \cdot 30)) = -13.1$$

BMI	Blood Pressure
19	120
25	135
23	125
30	140

$$b0_{new} = 82 - 0.001 \cdot (-13.1) = 82.013$$

Gradient Descent Example

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

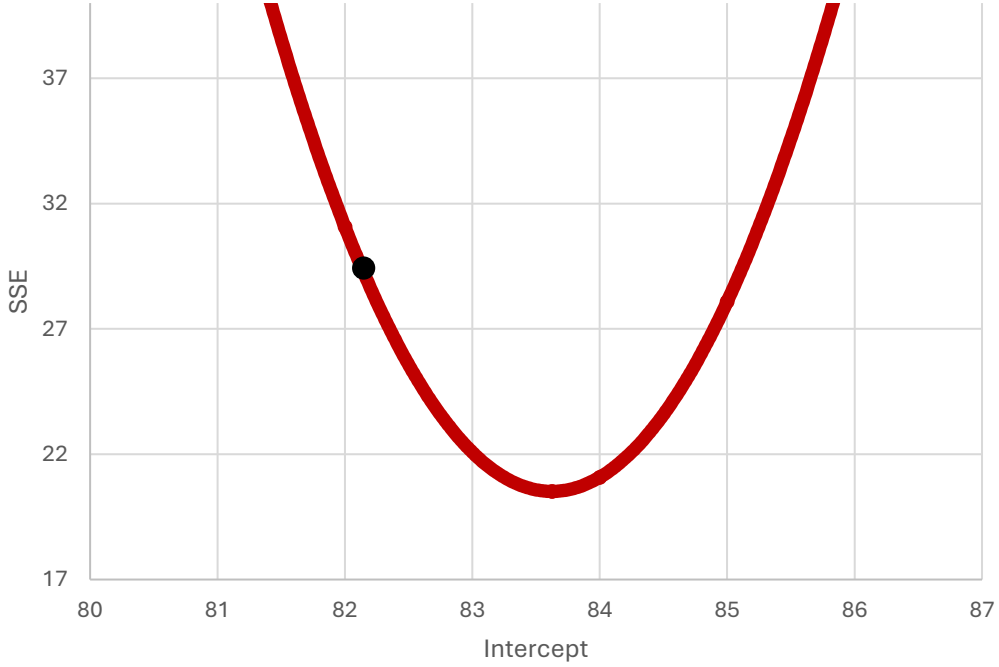
$$\gamma = 0.001$$

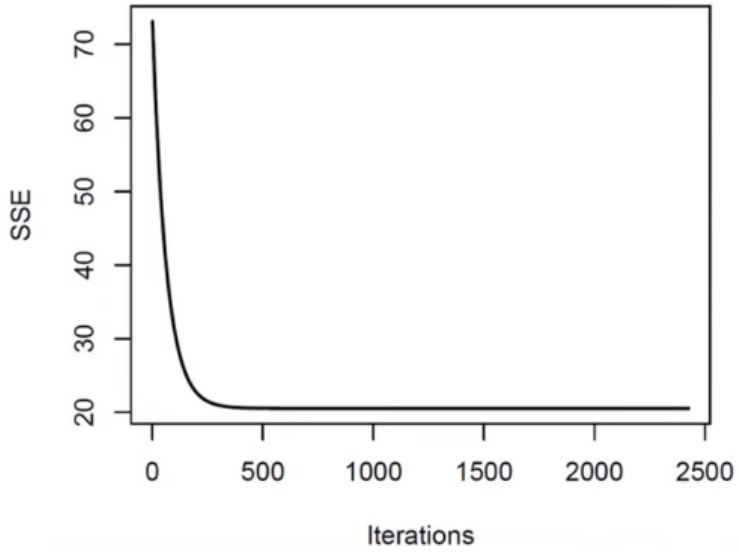
$$b_1 = 1.912$$

$$y = b_0 + b_1 x$$

$$b0_{new} = b0_{old} - \gamma \nabla f(b0_{old})$$

$$b0_{new} = 82.013 - 0.001 \nabla f(82.013)$$



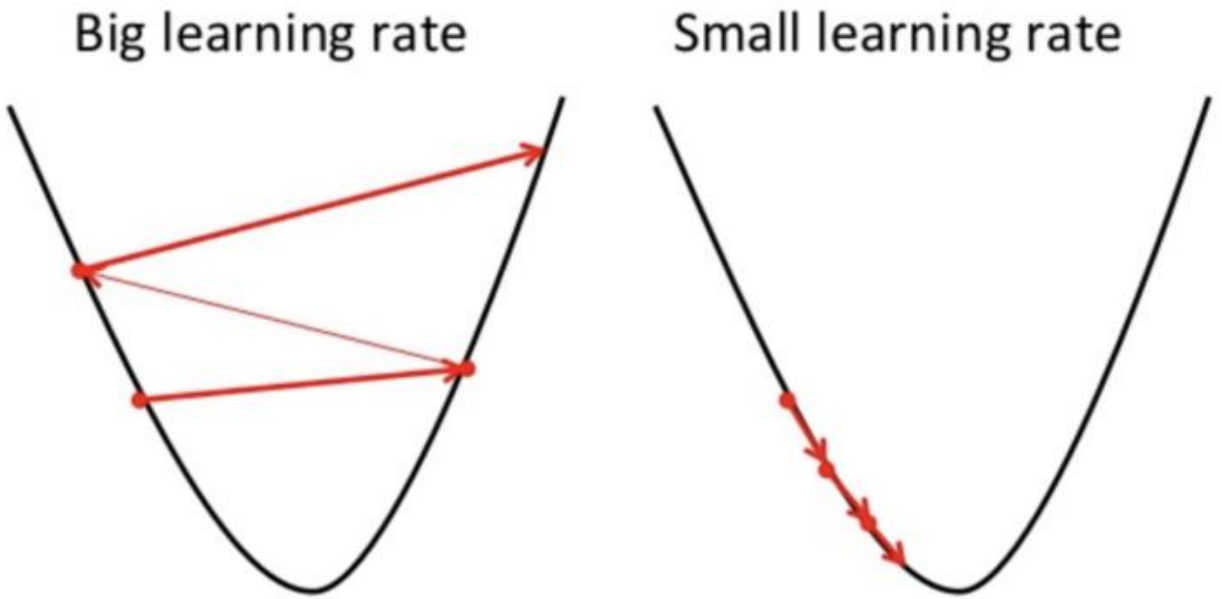


BMI	Blood Pressure
19	120
25	135
23	125
30	140

Challenges in Gradient Descent

Choice of Learning Rate

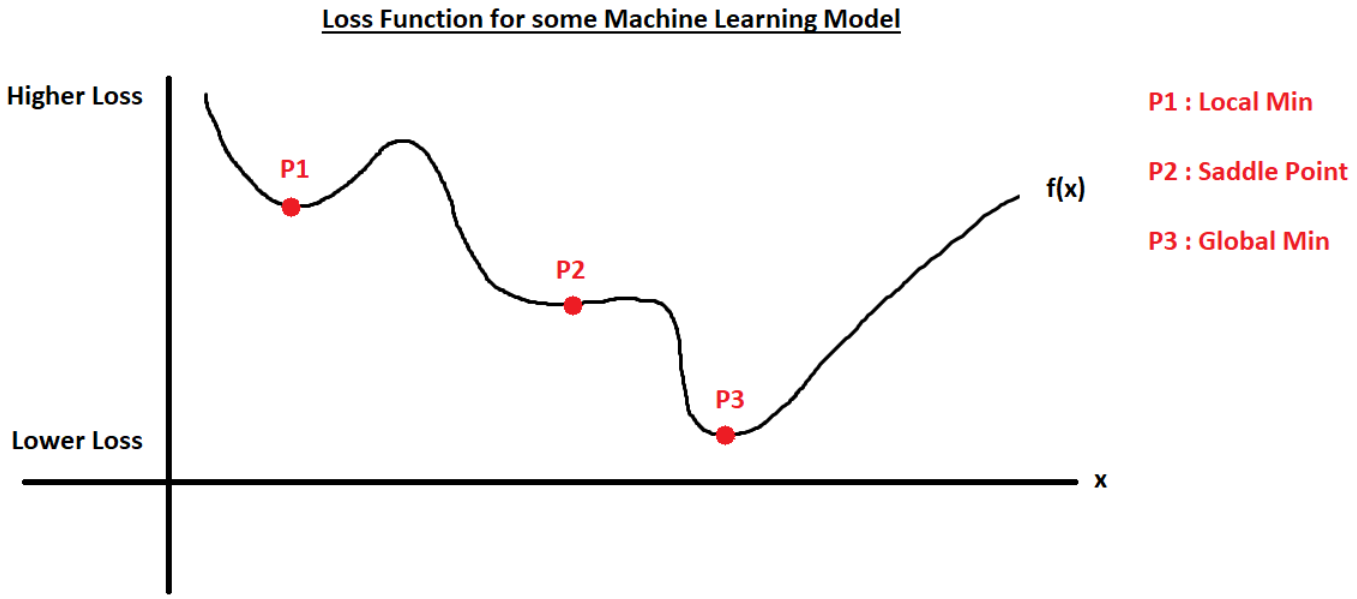
- **Too large**: Can cause overshooting and divergence.
- **Too small**: Results in slow convergence.



Challenges in Gradient Descent

Local Minima and Saddle

Can get stuck in local minima or flat regions (saddle points) of non-convex functions.



Other Alternatives

- Use **Stochastic Gradient Descent (SGD)** or **Mini-Batch Gradient Descent** to speed up convergence and deal with large datasets.
- **Stochastic Gradient Descent (SGD)** is a variation of the traditional **Gradient Descent** algorithm that updates the model parameters using only a single training example (or a small subset) at each step, rather than using the entire dataset. This makes it faster and more efficient, especially for large datasets.
- **Mini-batch gradient descent** is a variation of the gradient descent algorithm that splits the training dataset into small batches that are used to calculate model error and update model coefficients.

Stochastic Gradient Descent

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

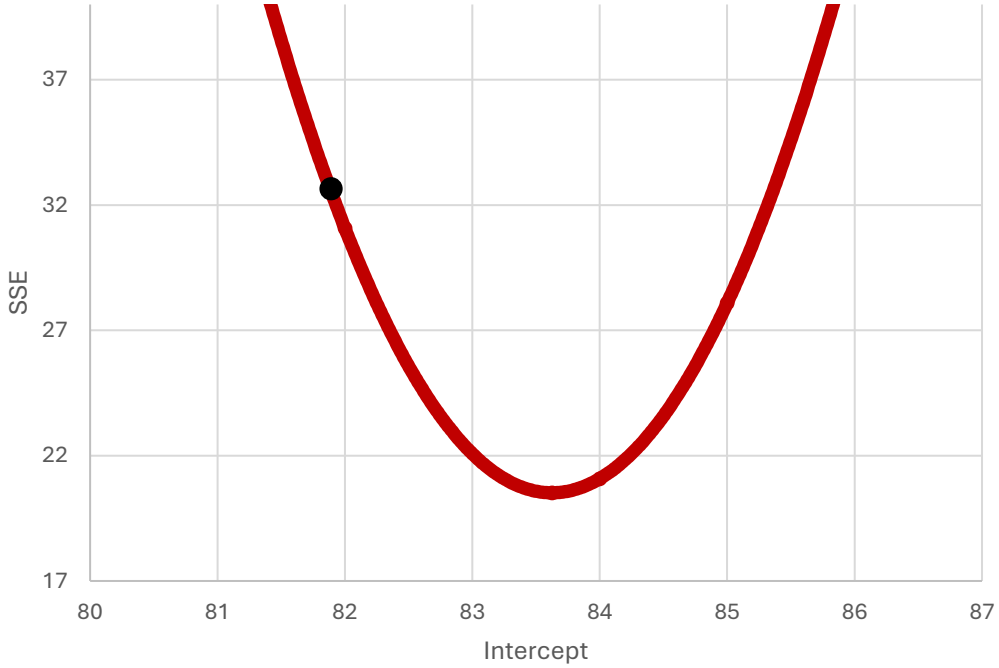
$$\gamma = 0.001$$

$$b_1 = 1.912$$

$$y = b_0 + b_1 x$$

$$\text{SSE} = \sum_{i}^{n} (y_i - (b_0 + b_1 x_i))^2$$

$$\frac{\partial \text{SSE}}{\partial b_0} = -2 \sum_{i}^{n} (y_i - (b_0 + b_1 x_i))$$



$$b0_{new} = b0_{old} - \gamma \nabla f(b0_{old})$$

$$b0_{new} = 82 - 0.001 \nabla f(82)$$

$$\frac{\partial \text{SSE}}{\partial b_0} = -2((120 - 82 - 1.912 \cdot 19) = -3.344$$

BMI	Blood Pressure
19	120
25	135
23	125
30	140

$$b0_{new} = 82 - 0.001 \cdot (-3.344) = 82.00334$$

$$\frac{\partial \text{SSE}}{\partial b_0} = -2((125 - 82.00334 - 1.912 \cdot 23) = 1.959$$

$b0_{new}$ = 82.00334-0.001 . (1.959)= 82.00138

Stochastic Gradient Descent

$$\text{SSE} = \sum_i^n (y_i - \hat{y}_i)^2$$

$$\gamma = 0.001$$

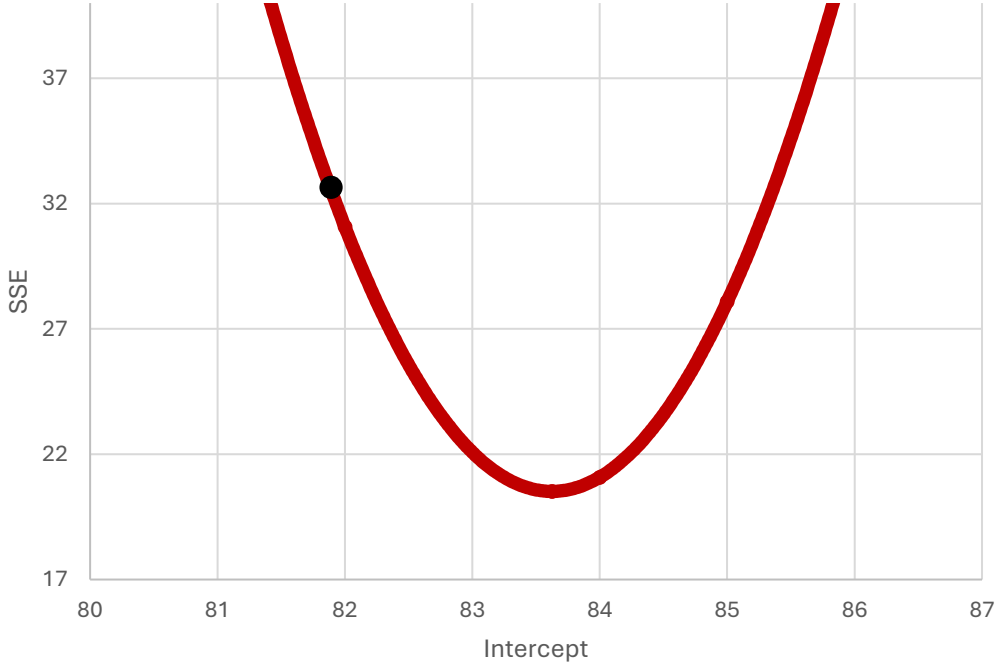
$$b_1 = 1.912$$

$$\text{SSE} = \sum_i^n (y_i - (b_0 + b_1 x_i))^2$$

$$\frac{\partial \text{SSE}}{\partial b_0} = -2 \sum_i^n (y_i - (b_0 + b_1 x_i))$$

$$y = b_0 + b_1 x$$

$$b0_{new} = b0_{old} - \gamma \nabla f(b0_{old})$$



$$b0_{new} = 82.00138 - 0.001 \nabla f(82.00138)$$

BMI	Blood Pressure
19	120
25	135
23	125
30	140

$$\frac{\partial \text{SSE}}{\partial b_0} = -2((140 - 82.00138 - 1.912 \cdot 30) = -1.2774$$

$b0_{new}$ = 82.00138- 0.001 . (-1.2774) = 82.00266

$$\frac{\partial \text{SSE}}{\partial b_0} = -2((135 - 82.00266 - 1.912 \cdot 25) = -10.395$$

$b0_{new}$ = 82.00266 – 0.001.(-10.395) = 82.013

Mini-Batch Gradient Descent

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

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

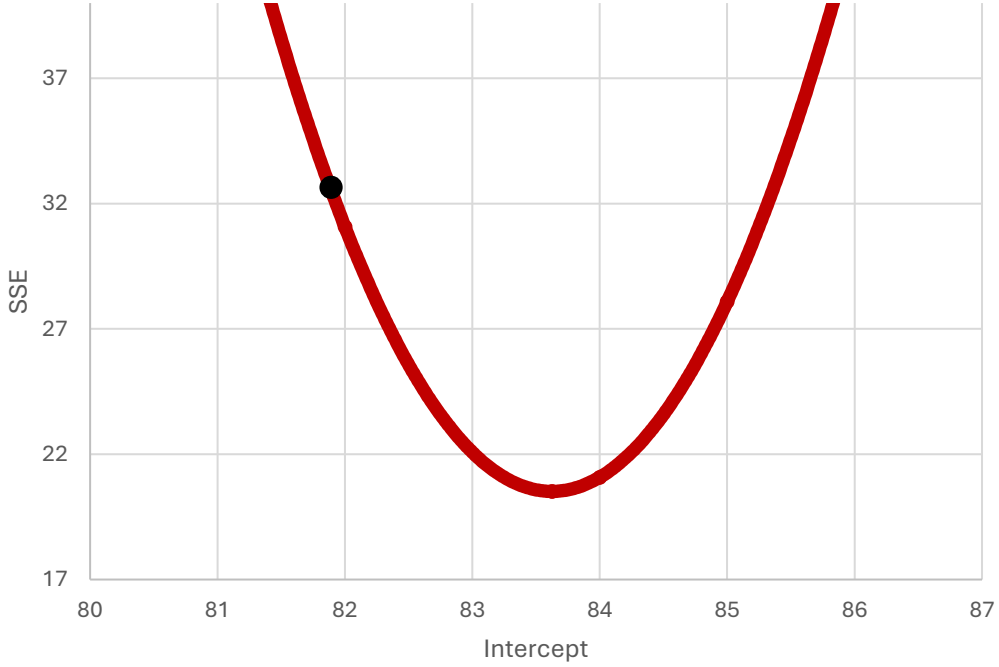
$$\gamma = 0.001$$

$$b_1 = 1.912$$

$$\text{SSE} = \sum_{i}^{n} (y_i - (b_0 + b_1 x_i))^2$$

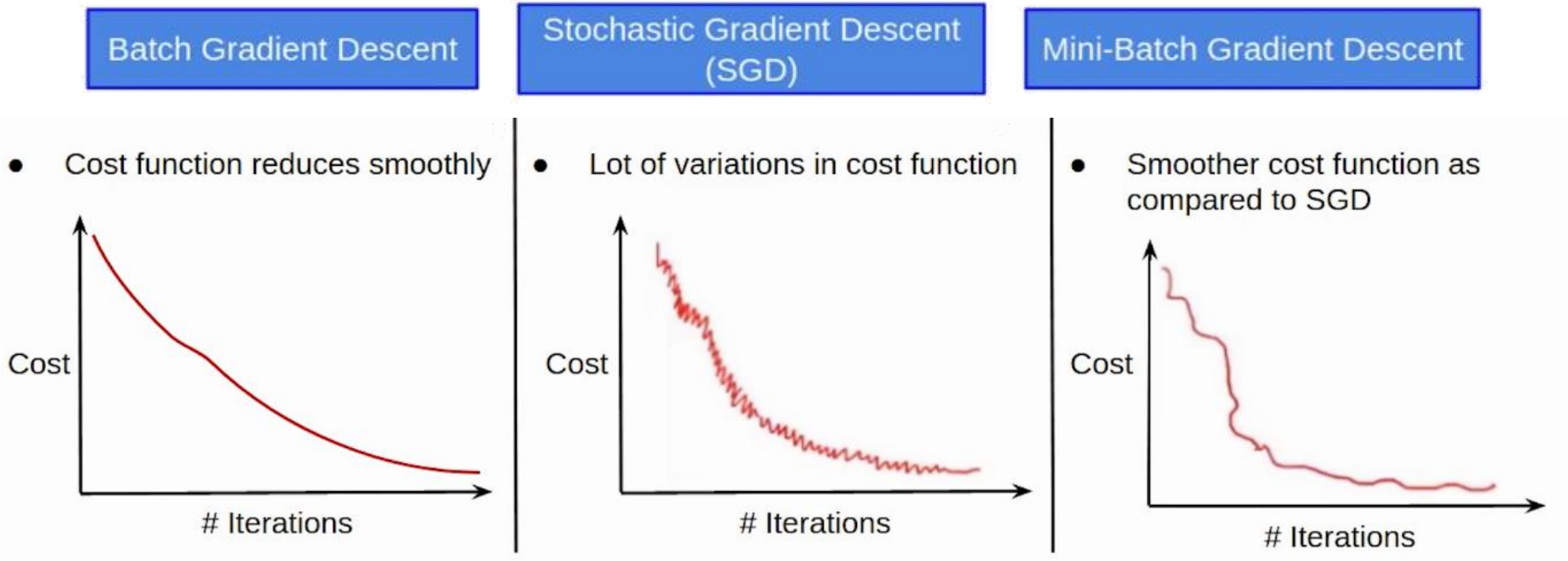
$$\frac{\partial \text{SSE}}{\partial b_0} = -2 \sum_{i}^{n} (y_i - (b_0 + b_1 x_i))$$

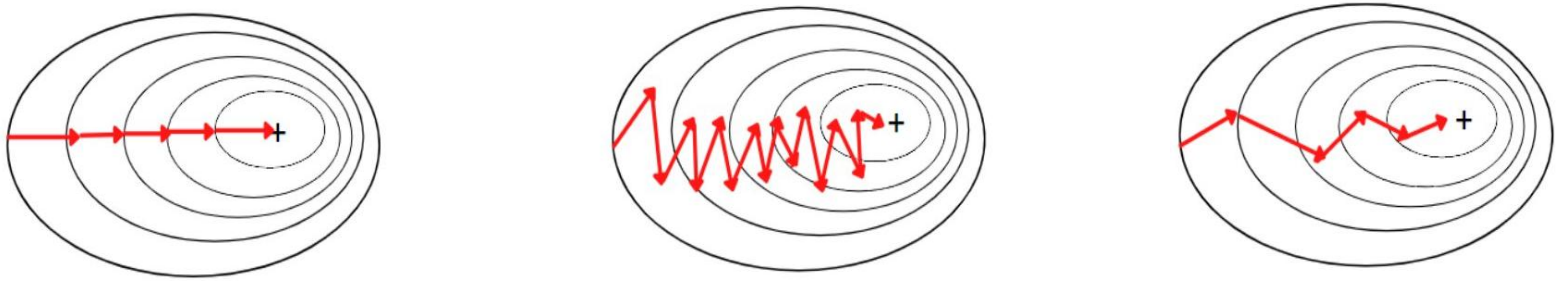
$$y = b_0 + b_1 x$$



BMI	Blood Pressure
19	120
25	135
23	125
30	140

Batch GD VS SGD VS Mini-batch GD





Batch GD VS SGD VS Mini-batch GD

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

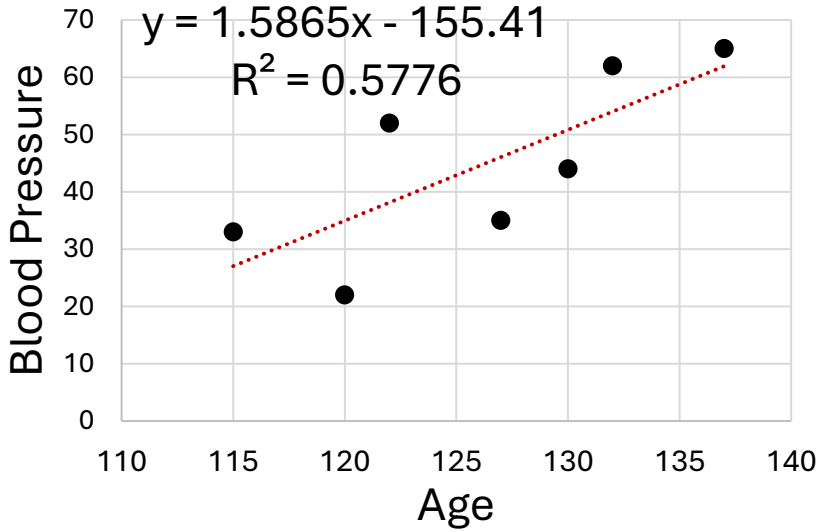
Aspect	Batch Gradient Descent	Stochastic Gradient Descent (SGD)	Mini-Batch Gradient Descent
Update Frequency	?????	?????	?????
Convergence Speed	?????	?????	?????
Pros	?????	?????	?????
Cons	?????	?????	?????
Memory Usage	?????	?????	?????
Suitability for Large Datasets	?????	?????	?????

Contents

Need for Regularization

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

BP	Age	Math Score
120	22	65
115	33	34
127	35	35
130	44	43
122	52	54
132	62	60
137	65	22

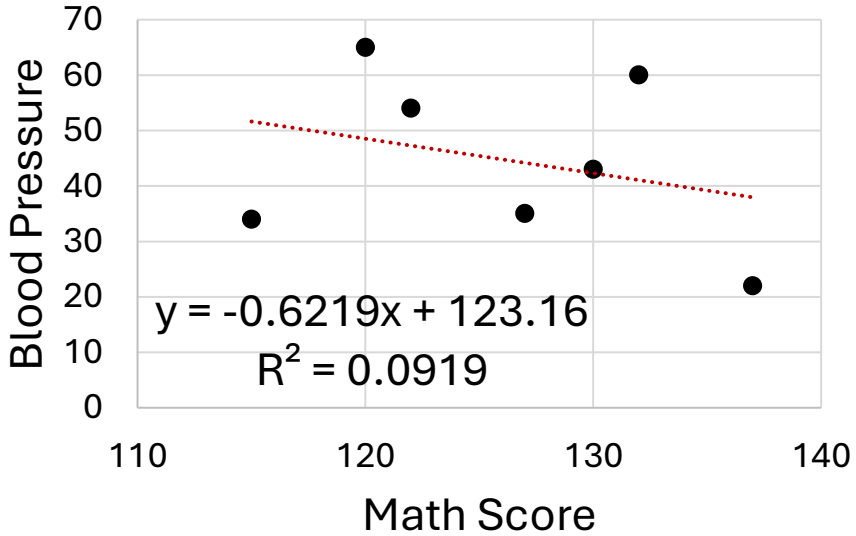


$$BP = \beta_0 + \beta_1 Age + \beta_2 Score$$

$R^2 = 0.578$ $\quad BP = \beta_0 + \beta_1 Age$

$R^2 = 0.092$ $\quad BP = \beta_0 + \beta_1 Score$

$R^2 = 0.592$ $\quad BP = \beta_0 + \beta_1 Age + \beta_2 Score$



$$BP = \beta_0 + \beta_1 Age + \beta_2 Score + \beta_3 Color + \beta_4 Car + ...$$

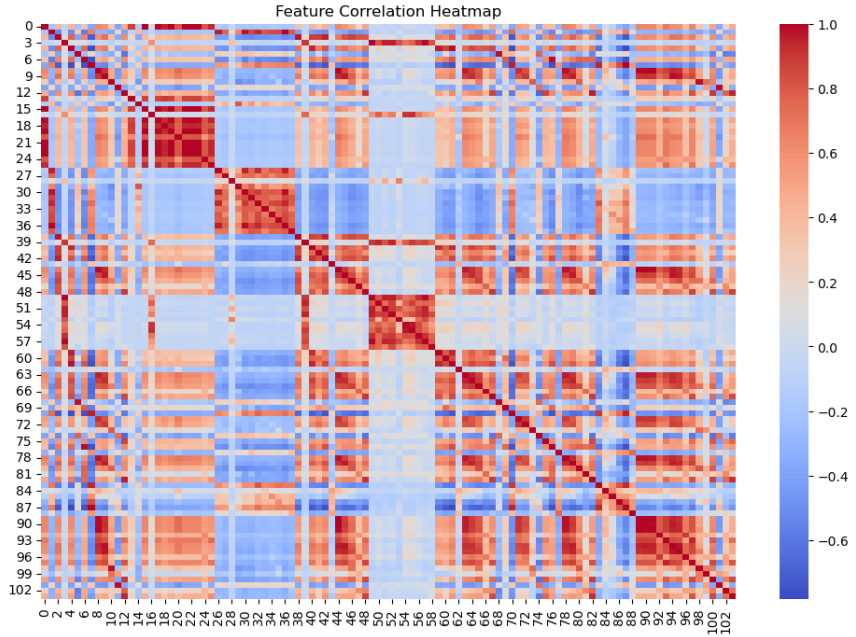
$R^2 = 0.999$

Need for Regularization

	Feature_0	Feature_1	Feature_2	Feature_3	Feature_4	Feature_5	Feature_6	Feature_7	Feature_8	Feature_9	...	Feature_95	Feature_96	Feature_97	Feature_98
0	0.000000	0.18	0.067815	0.0	0.314815	0.577505	0.641607	0.269203	0.000000	0.208015	...	0.059749	0.208015	0.018655	0.082503
1	0.000236	0.00	0.242302	0.0	0.172840	0.547998	0.782698	0.348962	0.043478	0.104962	...	0.058064	0.104962	0.021462	0.306021
2	0.000236	0.00	0.242302	0.0	0.172840	0.694386	0.599382	0.348962	0.043478	0.104962	...	0.058064	0.103885	0.006661	0.306021
3	0.000293	0.00	0.063050	0.0	0.150206	0.658555	0.441813	0.448545	0.086957	0.066794	...	0.043345	0.066412	0.002230	0.421118
4	0.000705	0.00	0.063050	0.0	0.150206	0.687105	0.528321	0.448545	0.086957	0.066794	...	0.043345	0.066794	0.006635	0.421118
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

```
Training set score: 0.95
Test set score: 0.61
```

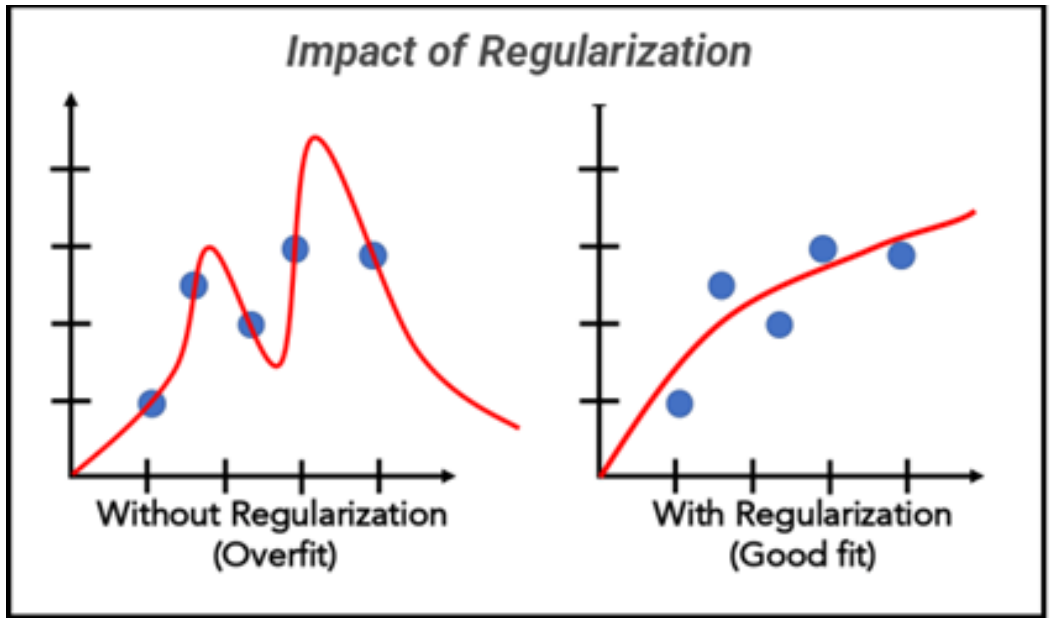
	Feature Index	Coefficient
0	0	-412.710947
1	1	-52.243207
2	2	-131.898815
3	3	-12.004136
4	4	-15.510713



Large coefficients signal overfitting
Test score is much lower than training score

Need for Regularization

- **Regularization** is a crucial technique in machine learning and statistical modeling used to **prevent overfitting** and improve the generalization of a model.
- In a mathematical or ML context, we make something regular by adding information which creates a solution that prevents overfitting.
- The “something” we’re making regular in our ML context is the “objective function”, something we try to minimize during the optimization problem.



Lasso Regression

- Called L1 regularization
- Will cause many weights to be exactly 0
- parameter α to control the strength of regularization.
- Will have a 'sweet spot' depending on the data
- No closed-form solution, but no longer strictly convex, and not differentiable
- Weights can be optimized using *coordinate descent*

$$J(w) = \sum_{i=1}^{n}(y_i - (w_0 + \sum_{j=1}^{p} w_j x_{ij}))^2 + \lambda \sum_{j=1}^{p} |w_j|$$

$\hat{y}$ = Predicted value

w_0 = Intercept (bias term)

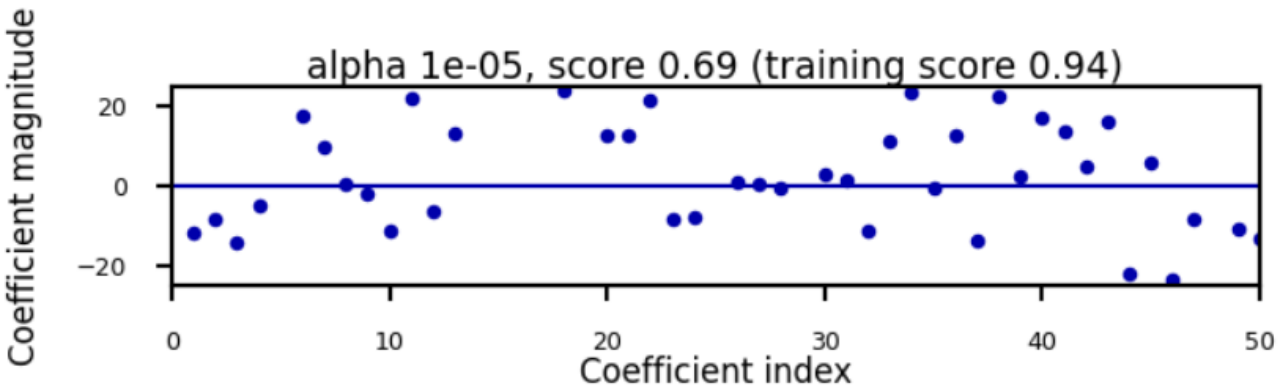
w_j = Coefficients (weights) for each feature x_j

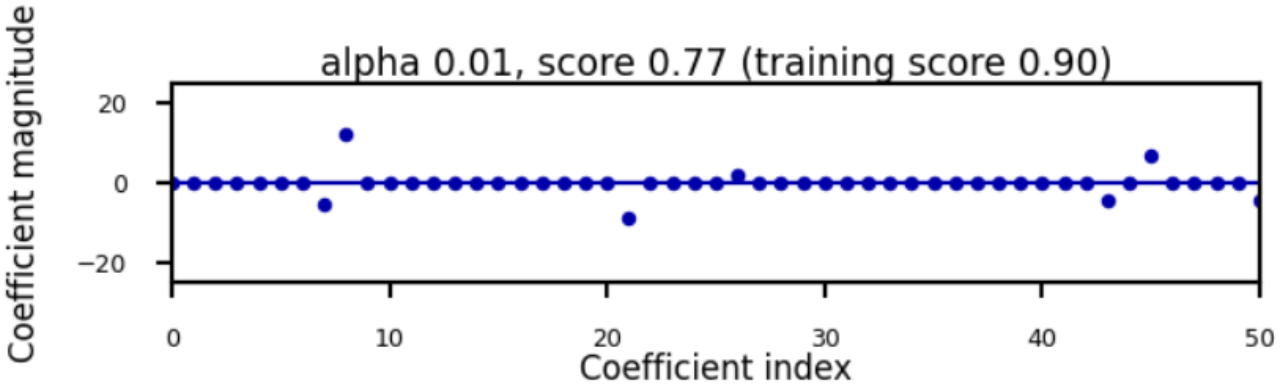
x_j = Feature values

p = Number of features

Lasso Regression

- L1 prefers coefficients to be exactly zero (sparse models)
- Some features are ignored entirely: automatic feature selection





Ridge Regression

- Model is penalized if it uses large coefficients (w)
- Each feature should have as little effect on the outcome as possible
- We don't want to penalize w_0 , so we leave it out
- Regularization: explicitly restrict a model to avoid overfitting.
- Called L2 regularization because it uses the L2 norm: $\sum w_i^2$

$$J(w) = \sum_{i=1}^{n}(y_i - (w_0 + \sum_{j=1}^{p} w_j x_{ij}))^2 + \lambda \sum_{j=1}^{p} w_j^2$$

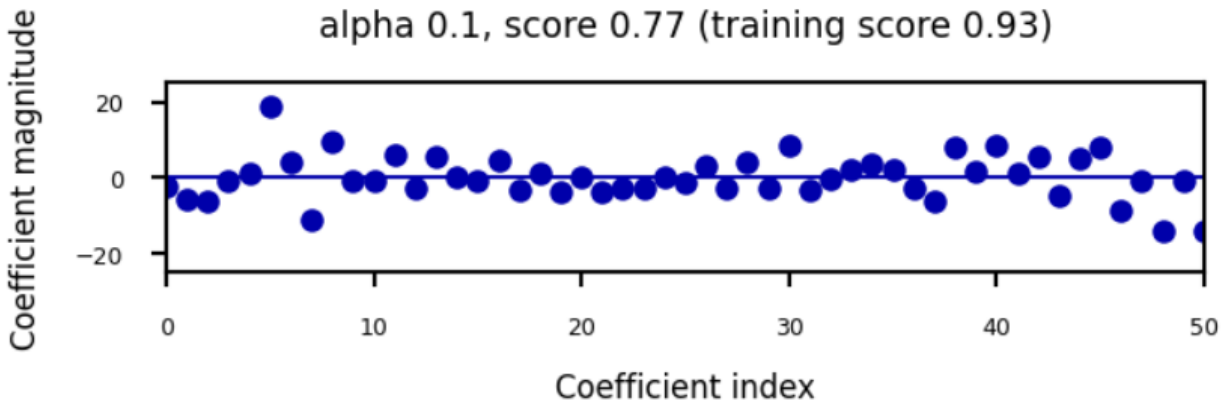
w_0 = Intercept (bias term)

w_j = Coefficients (weights) for each feature x_j

x_j = Feature values

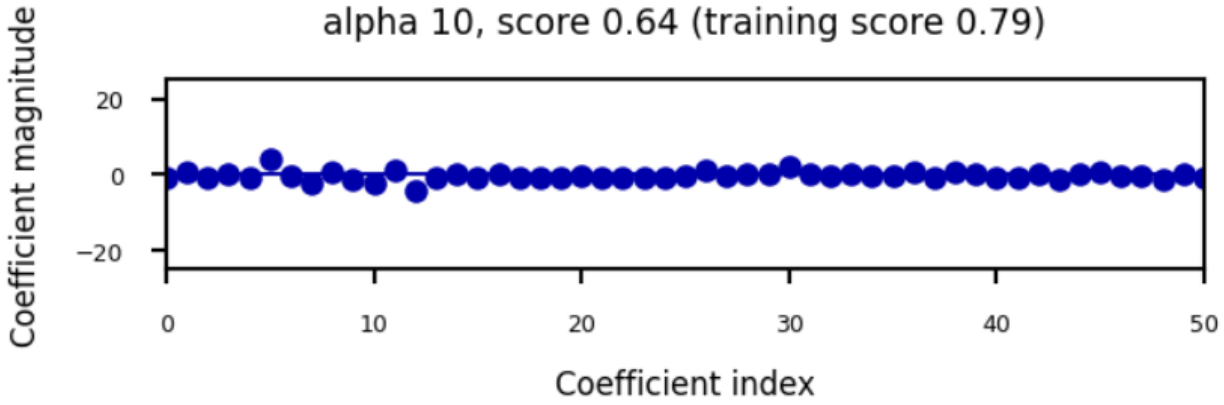
p = Number of features

- .Increasing regularization decreases the values of the coefficients, but never to 0



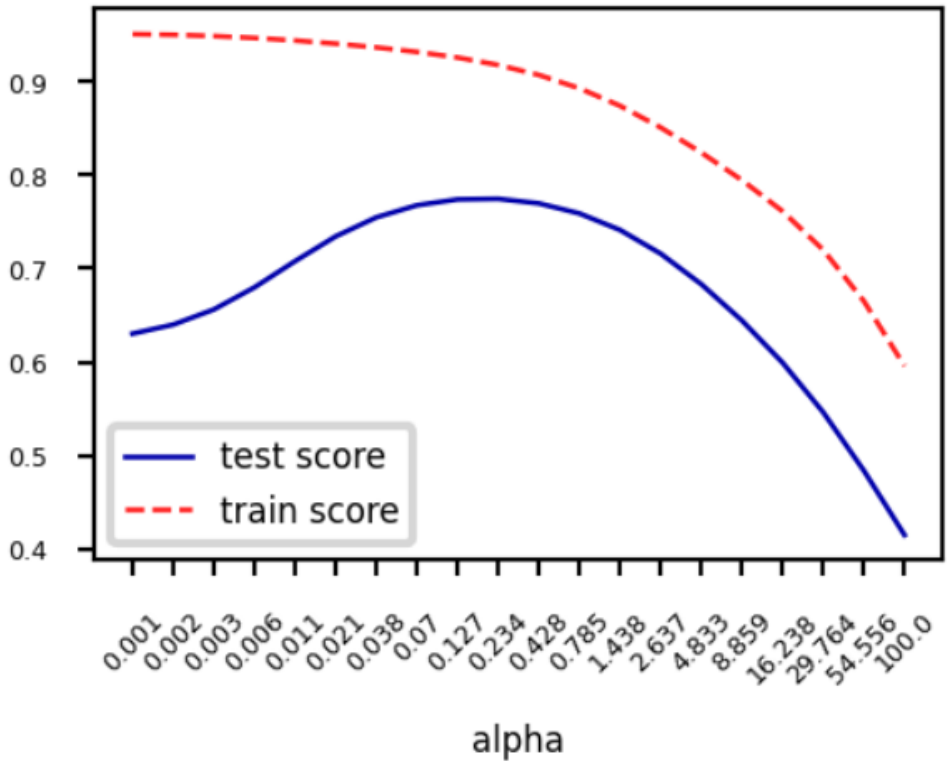
Test set score is higher and training set score lower: less overfitting!

```
Training set score: 0.89
Test set score: 0.75
```



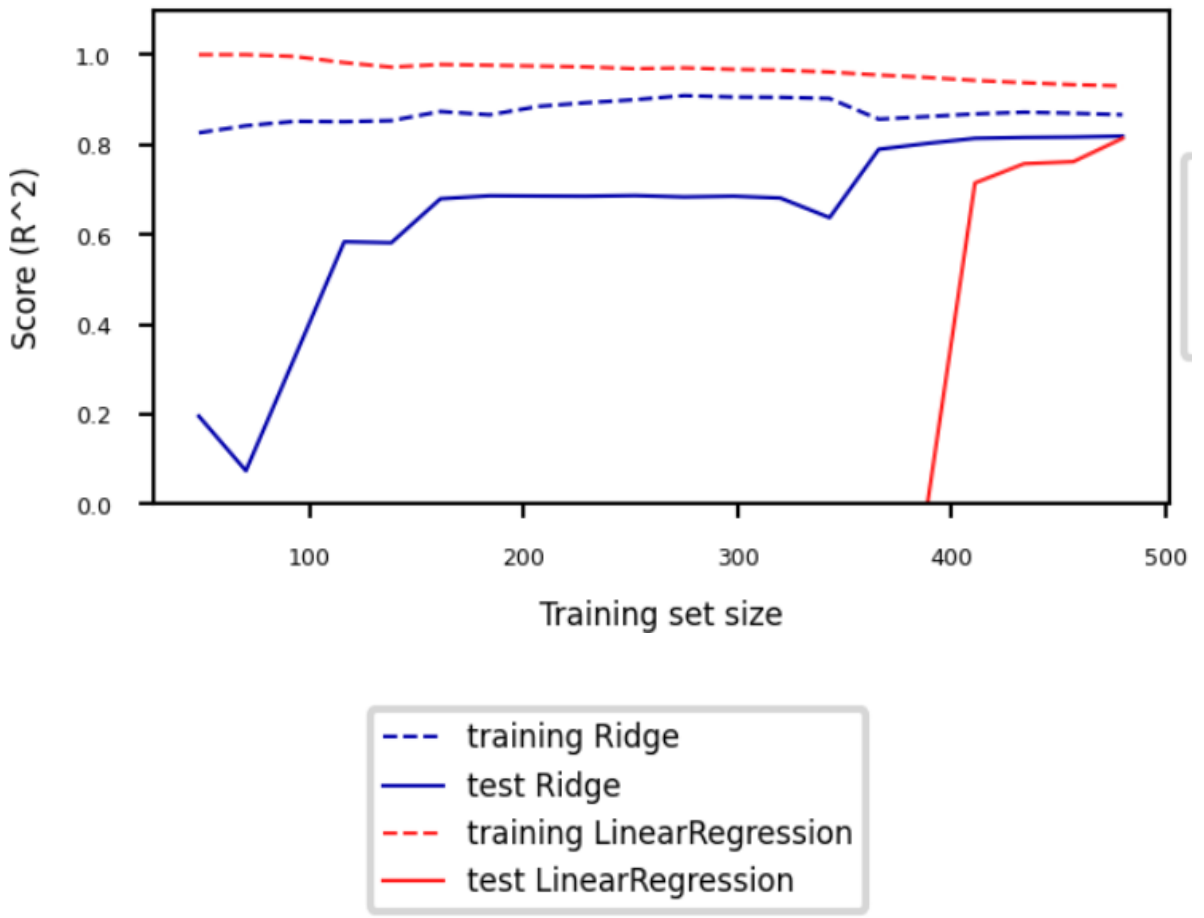
Hyper tuning of α

- When we plot the train and test scores for every α value, we see a sweet spot Models with smaller α are overfitting
  - Models with larger α are underfitting



Other ways to reduce overfitting

- Add more training data: with enough training data, regularization becomes less important
  - Ridge and ordinary least squares will have the same performance
- Use fewer features: remove unimportant ones or find a low-dimensional embedding (e.g. PCA)
  - Fewer coefficients to learn, reduces the flexibility of the model
- Scaling the data typically helps (and changes the optimal αα value)



Elastic Net

- Elastic Net combines **Lasso (L1) and Ridge (L2) regression** into a single model. It **balances feature selection (Lasso) and coefficient shrinkage (Ridge).**
- If **l1_ratio = 1** → Pure **Lasso**.
- If **l1_ratio = 0** → Pure **Ridge**.
- If **0 < l1_ratio < 1** → Mix of Lasso and Ridge.

$$J(w) = \sum_{i=1}^{n}(y_i - (w_0 + \sum_{j=1}^{p} w_j x_{ij}))^2 + \lambda_1 \sum_{j=1}^{p} |w_j| + \lambda_2 \sum_{j=1}^{p} w_j^2$$

w_0 = Intercept (bias term)

w_j = Coefficients (weights) for each feature x_j

x_j = Feature values

p = Number of features

Aspect	Linear Regression	Lasso	Ridge	Elastic Net
Regularization	?????	?????	?????	?????
Penalization	?????	?????	?????	?????
Feature Selection	?????	?????	?????	?????
Use Case	?????	?????	?????	?????
Computation	?????	?????	?????	?????

Summary

1. Explain the concept and cost function of linear regression.
2. Evaluate model performance using metrics like Mean Squared Error (MSE).
3. Understand and implement gradient descent to minimize the cost function.
4. Differentiate between Mini-Batch and Stochastic Gradient Descent (SGD).
5. Apply L1 (Lasso) and L2 (Ridge) and ElasticNet regularization to prevent overfitting.

Quiz

https://forms.office.com/r/Ssde21R4Qi