5CS037-Concepts and Technologies of AI

Lecture-07

Gentle Introduction towards Supervised Machine Learning

Linear Methods for Regression.

Siman Giri

Story So Far…….

From Last Week: Components of Machine Learning..

- **Dataset:**
  - Labelled vs. Unlabeled Dataset.
- **A Decision Process (Representation/Model):**
  - Machine learning algorithms(Models) are used to make inference or estimate of an output based on input data – labeled or unlabeled.
- **An Error Function (Evaluation):**
  - A performance metric used to evaluate the estimate of a model.
  - Metrics depends on types of learning (supervised or unsupervised) and types of task (Classification or Regression)
- **An model Optimization Process:**
  - An automated algorithm or process used to update parameters of machine learning models until threshold or accepted evaluation metric has been achieved

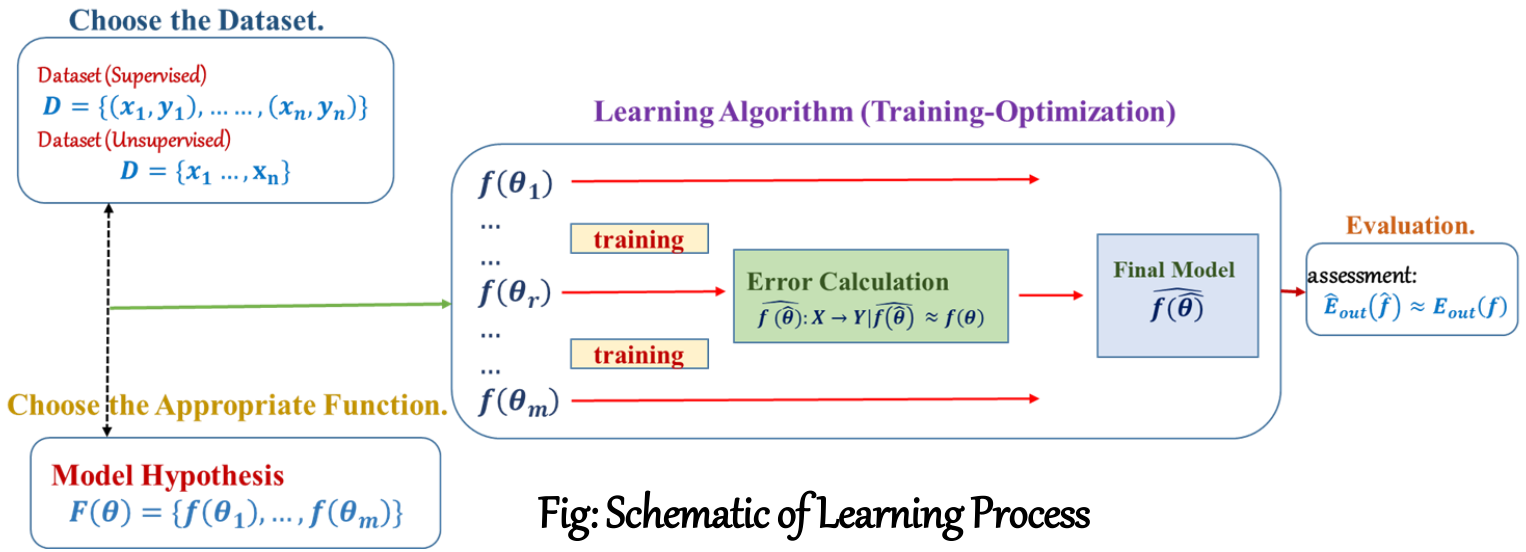


Fig: Schematic of Learning Process

From Last Week: Supervised Machine Learning

- It is an attempt to find the function *"f"* that minimizes the selected loss such that:
  - $f(\theta) = argmin_{h \in H} \mathbb{L}(h)$ **{θ: a function parameter to be learned}**
- A big part of machine learning focuses on the question, how to do this minimization efficiently?
- If you find a function ***f(.)*** with low loss on your data ***D***, how do you know whether it will still get examples right that are not in ***D***?
  - **Generalization!!!**

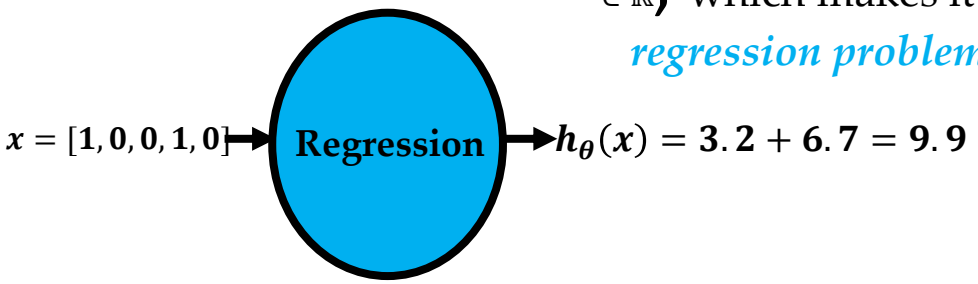
From Last Week: Regression Vs. Classification.

- **Regression:**
  - What are the possible outputs of the linear regression function

$$Y = \hat{f}_\theta(x) \Rightarrow ?$$

Real-valued Outputs.

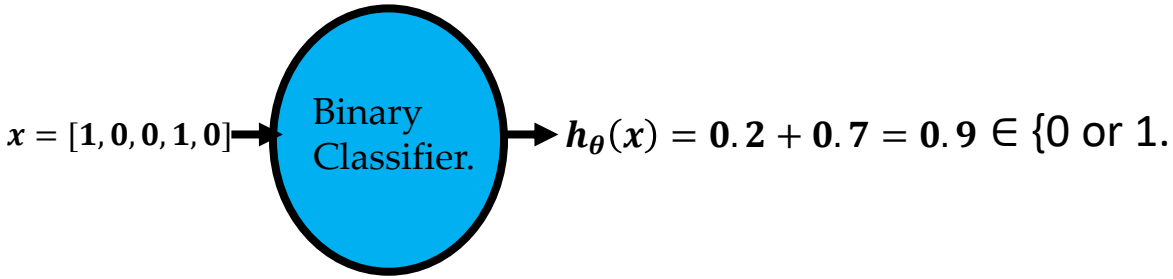
$\in \mathbb{R}$, which makes it a *regression problem*



- **Classification:**
  - What are the possible outputs of the linear regression function

$$Y = \hat{f}_\theta(x) \Rightarrow ?$$

Discrete Outputs.



1. Linear Models for Regression.

Linear Regression

1.1 Linear Regression: Data and Motivation.

Head Size(cm^3)	Brain Weight(grams)
4512	1530
3738	1297
4261	1335
3777	1282
4177	1590
3585	1300
3785	1400
3559	1255
3613	1355
3982	1375
3443	1340
3993	1380
3640	1355
4208	1522
3832	1208
3876	1405
3497	1358
3466	1292
3095	1340
4424	1400
3878	1357

Task(Q): What is the Brain Weight for, head size of 3000 cm^3?

Let's analyze the data to figure out the relationship:

feature $\{X\}$ ➔ target $\{Y\}$.

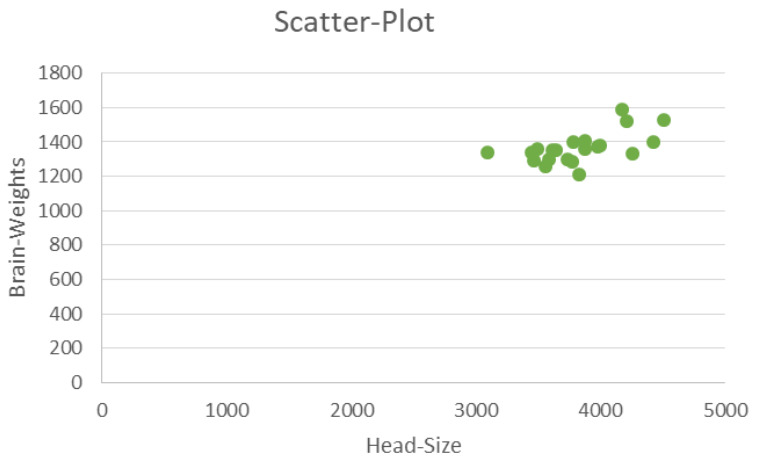
1.1 Linear Regression: Data and Motivation.

Head Size(cm^3)	Brain Weight(grams)
4512	1530
3738	1297
4261	1335
3777	1282
4177	1590
3585	1300
3785	1400
3559	1255
3613	1355
3982	1375
3443	1340
3993	1380
3640	1355
4208	1522
3832	1208
3876	1405
3497	1358
3466	1292
3095	1340
4424	1400
3878	1357

Task(Q): What is the Brain Weight for, head size of 3000 cm^3?

Let's analyze the data to figure out the relationship:
feature $\{X\}$ ➔ target $\{Y\}$.

Data Exploration: Let's plot the Data:



What can you Observe?

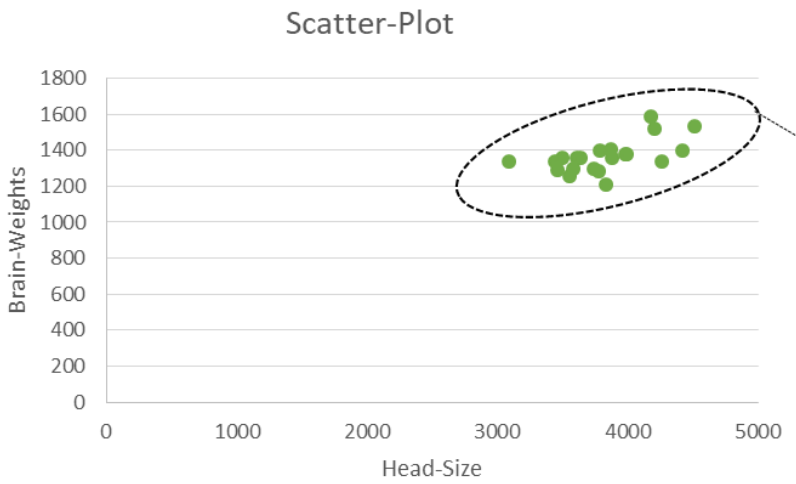
1.1 Linear Regression: Data and Motivation.

Head Size(cm^3)	Brain Weight(grams)
4512	1530
3738	1297
4261	1335
3777	1282
4177	1590
3585	1300
3785	1400
3559	1255
3613	1355
3982	1375
3443	1340
3993	1380
3640	1355
4208	1522
3832	1208
3876	1405
3497	1358
3466	1292
3095	1340
4424	1400
3878	1357

Task(Q): What is the Brain Weight for, head size of 3000 cm^3?

Let's analyze the data to figure out the relationship:
feature $\{X\}$ ➔ target $\{Y\}$.

Data Exploration: Let's plot the Data:



Some Patterns Seems to be appearing.

What can you Observe?

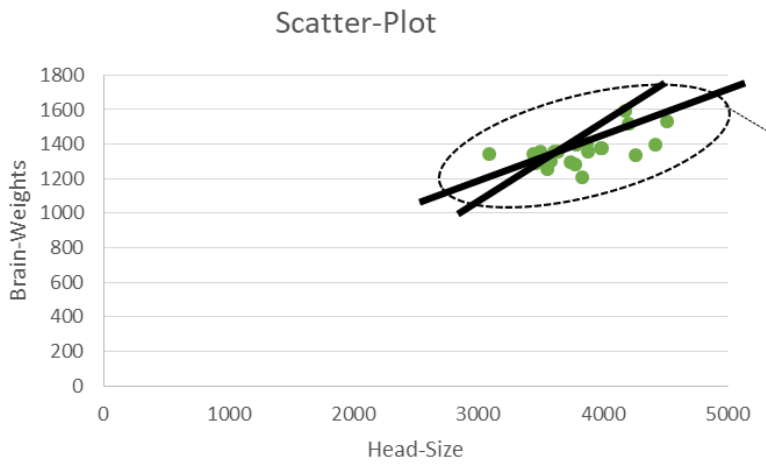
1.1 Linear Regression: Data and Motivation.

Head Size(cm^3)	Brain Weight(grams)
4512	1530
3738	1297
4261	1335
3777	1282
4177	1590
3585	1300
3785	1400
3559	1255
3613	1355
3982	1375
3443	1340
3993	1380
3640	1355
4208	1522
3832	1208
3876	1405
3497	1358
3466	1292
3095	1340
4424	1400
3878	1357

Task(Q): What is the Brain Weight for, head size of 3000 cm^3?

Let's analyze the data to figure out the relationship:
feature $\{X\}$ ➔ target $\{Y\}$.

Data Exploration: Let's plot the Data:



Some Patterns Seems to be appearing.

What can you Observe?

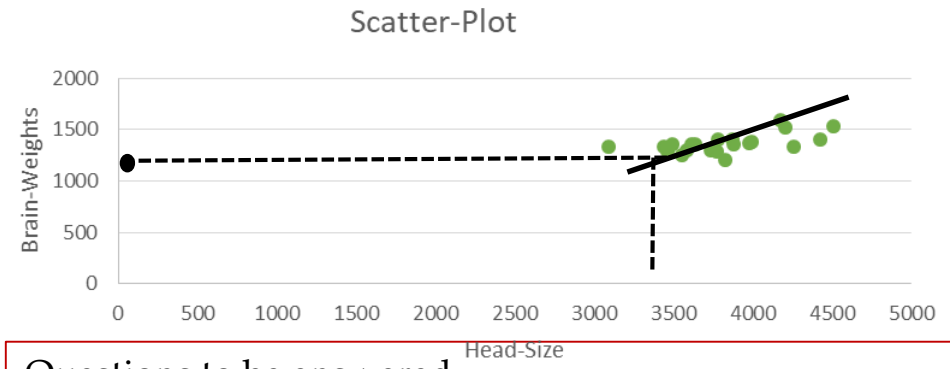
1.1 Linear Regression: Data and Motivation.

Head Size(cm^3)	Brain Weight(grams)
4512	1530
3738	1297
4261	1335
3777	1282
4177	1590
3585	1300
3785	1400
3559	1255
3613	1355
3982	1375
3443	1340
3993	1380
3640	1355
4208	1522
3832	1208
3876	1405
3497	1358
3466	1292
3095	1340
4424	1400
3878	1357

Task(Q): What is the Brain Weight for, head size of 3000 cm^3?

Let's analyze the data to figure out the relationship:
feature $\{X\}$ ➔ target $\{Y\}$.

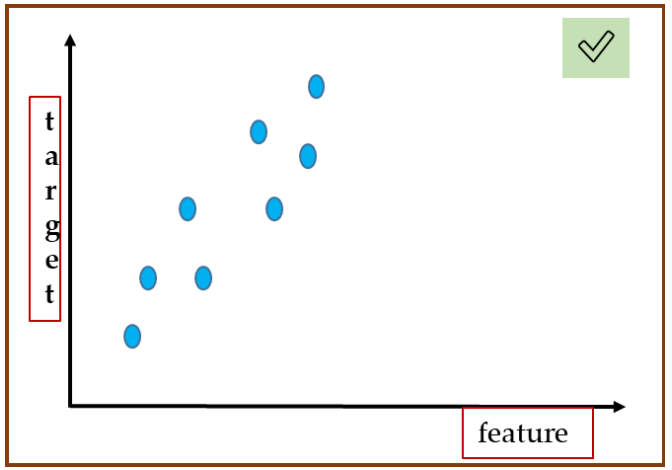
If we can learn a numerical model i.e. equation of a line, we can use the model to infer any target value, given feature value.

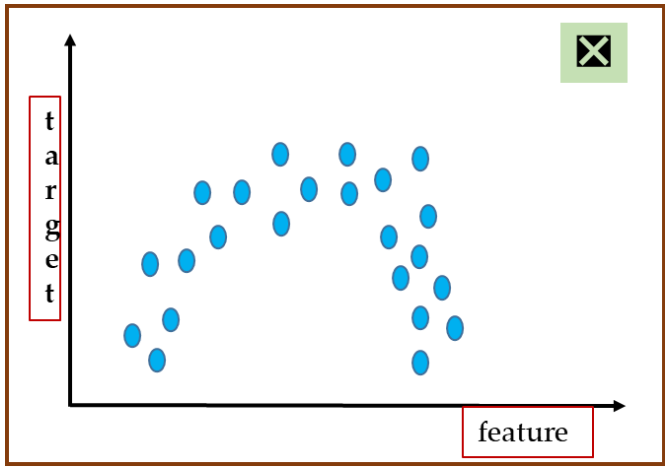


Questions to be answered:
How to learn a (mathematical)models?
How to pick best model(line) among all the possible lines?

1.2 Linear Regression: Introduction.

- Regression analysis is a tool to **investigate how two or more variables are related.**
- Linear regression attempts to **model the relationship between** two variables by **fitting a linear equation** to observed data.
  - One variable is considered to be an **explanatory(independent-feature) variable** $\{X\}$, and the other is considered to be a **dependent variable (target)** $\{Y\}$.
  - For example, In previous slides **we wanted to infer the weights of brain in neonates** to their **head size** using **linear regression model**.
- **Cautions !!!! :**
  - Before attempting to **fit a linear model to observed data**, a modeler should first determine **whether or not there is a relationship** between the **variables of interest**.
  - If there appears to be **no association** between the proposed explanatory and dependent variables (i.e., the scatterplot does not indicate any increasing or decreasing trends), then **fitting a linear regression model** to the data probably **will not provide a useful model**.

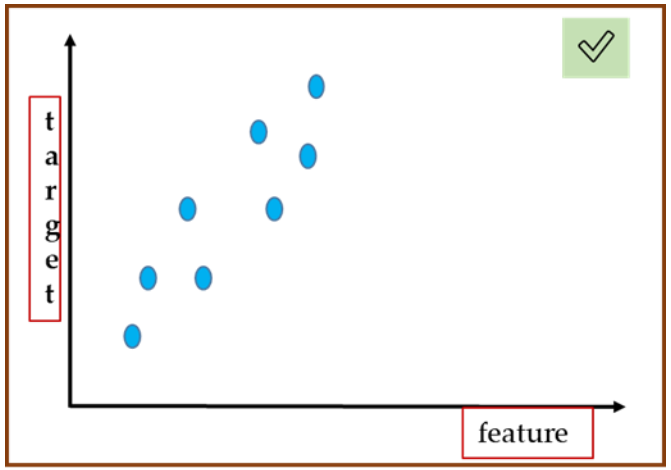




Mathematical Formulation of Linear Regression.

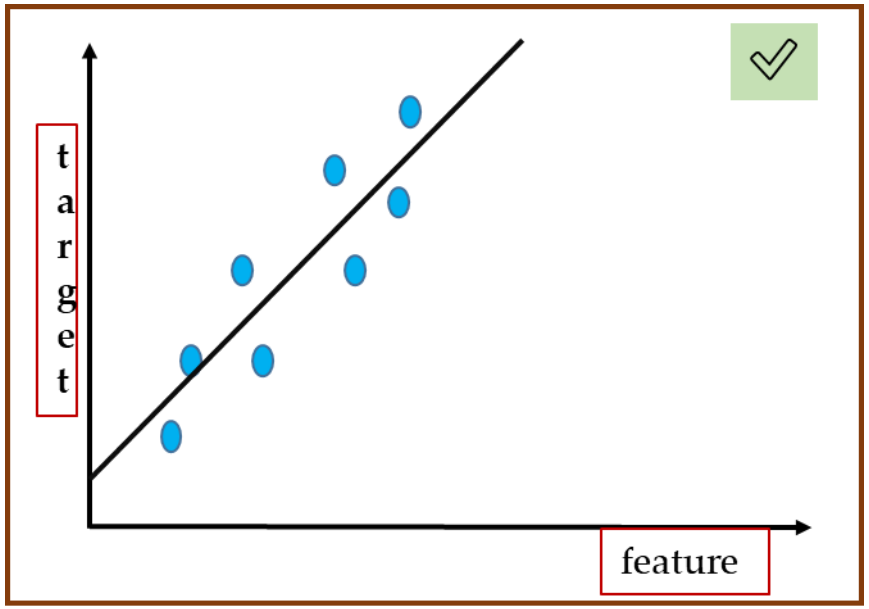
2. Component1: A Decision Process.

Q: What is the function that best fits these points?



2.1 Decision Process: Linear Regression.

- To define a useful model, we investigated the relationship between the response and predictor variables.
- Upon a close analysis we concluded **there exists a linear relationship**, which can be **represented using equation of straight line**, recall that equation of straight line has the following form:
  - $y = mx + b$
  - where **m** and **b** are the parameters of our linear function which represents:
    - **m: slope**
    - **b: y-intercept**
- **Question to answered:**
  - **[Q:1] How can we learn such a line?**
    - If we look closer towards the linear relationship between two variables, it is rarely the case where the coordinates fall exactly on a straight line.



2.2 Decision Process: Linear Regression.

- Definition:
- In the context of supervised machine learning we can define algorithm linear regression as:

Linear Regression:
It is a Supervised machine learning algorithm **that learns** a **dependent variable**, y, as a function of some **independent variable(s)(aka "features")**, x_i, by finding a **line** (surface) that **best fits** the **data**.
In general equation for linear regression model is written as:

$$y = w_0 + w_1 x_1 + \cdots + w_d x_d + \epsilon \text{ \{Multiple Linear Regression\}}$$
$$y = w_0 + w_1 x_1 + \epsilon \text{ \{Simple Linear Regression\}}$$

Where:
y: **the dependent variable: the thing we are trying to predict.**
x_i: **the independent variable: the features our model uses to model y.**
w_i: **the coefficients or parameters (aka "weights") of our regression model.**
ϵ: **the accumulated error (irreducible) in our model.**

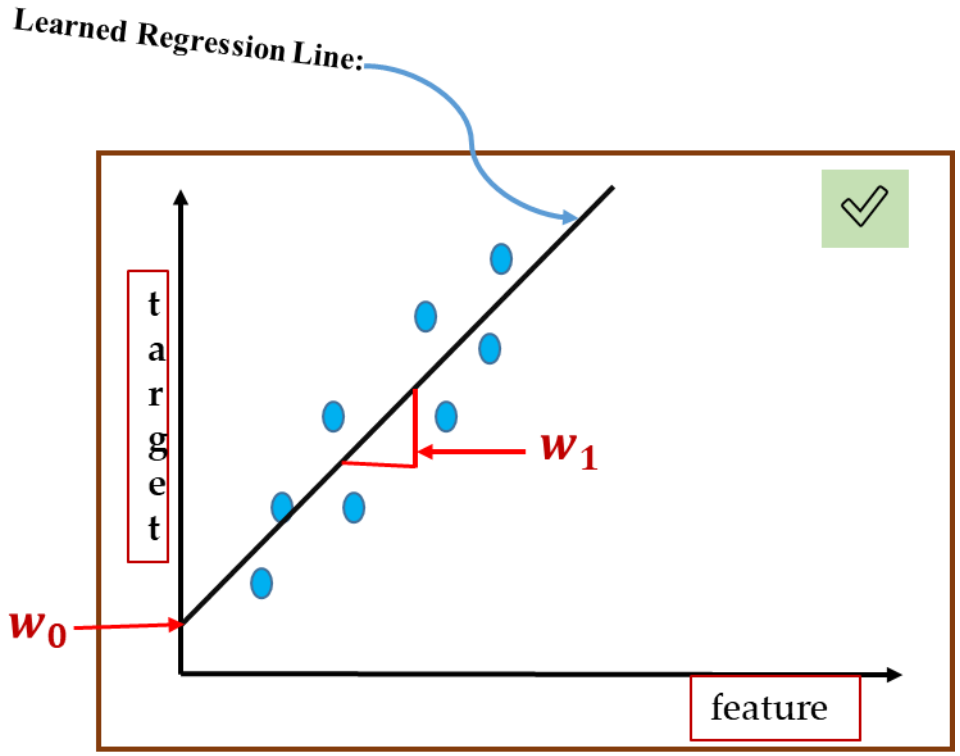
2.2 {Simple Vs. Multiple} Linear Regression

- aka ~ Univariate Linear Regression.
- One Dependent (Y) and One Independent Variable (X).
  - Mathematical Representation:
    - $y = w_0 + w_1 x_1 + \epsilon$
    - y: the dependent variable: the thing we are trying to predict.
    - x_i: the independent variable: the features our model uses to model y.
    - w_i: the coefficients or parameters (aka "weights") of our regression model.
    - ϵ: the accumulated error (irreducible) in our model.

- aka ~ Multivariate Linear Regression
- One Dependent Variable (Y) and Many Independent Variables (X).
  - Mathematical Representation:
    - $y = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_d x_d + \epsilon$
    - y: the dependent variable: the thing we are trying to predict.
    - x_i: the independent variable: the features our model uses to model y.
    - w_i: the coefficients or parameters (aka "weights") of our regression model.
    - ϵ: the accumulated error (irreducible) in our model.

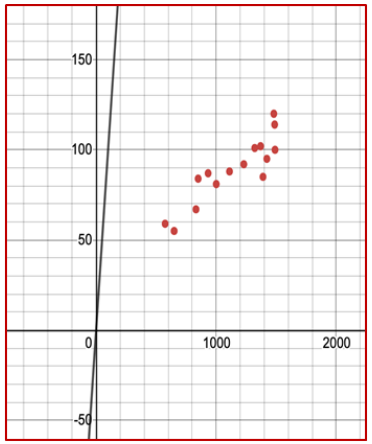
2.3 Linear Regression Function: Parameters.

- Given: functional representation of (simple) linear regression model:
  - $\hat{y} = w_0 + w_1 x_1 + \epsilon$
  - Parameter (w_0) : **biases** aka **intercepts**.
    - Represents $\hat{y}$: when $x_1 = 0$.
  - Parameter (w_1) : **weights** aka **slopes**.
    - Represents: rate of change i.e. **amount of changes** for x to get a **unit change** in y.
    - positive slope: one increases the other increases
    - negative slope: one increases the other decreases
    - zero slope: one increase other remains constant.

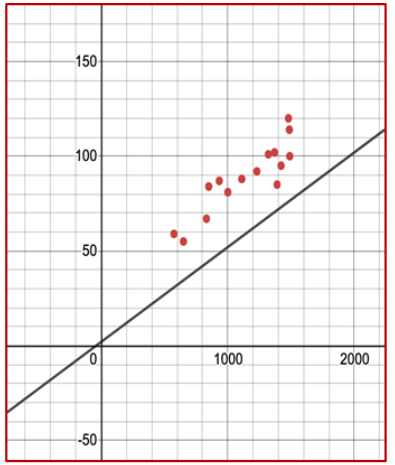


2.4 Decision Process: Linear Regression - Questions

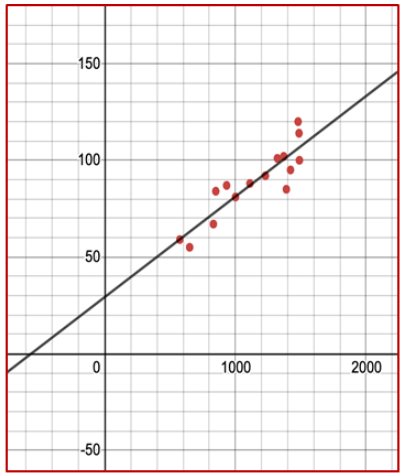
- Idea:
  - **Find a line that best represents (fits) our data.**
- Questions:
  - **[1]: How do we learn such line?**
    - Estimate ~ parameters (weights and biases) that best fits our data.
    - From a collection of functions [F: $\left\{f^{1}_{w_{0_1},w_{1_1}}, \dots, f^{k}_{w_{0_k},w_{1_k}}\right\}$] pick best function (best fit line).
  - **[2]: How do we know which line is best?**
    - **Any line which captures the most variability in the original data.**



$w_{1_1} = 1 \; and \; w_{0_1} = 0$



$w_{1_2} = 0.05 \; and \; w_{0_2} = 2$



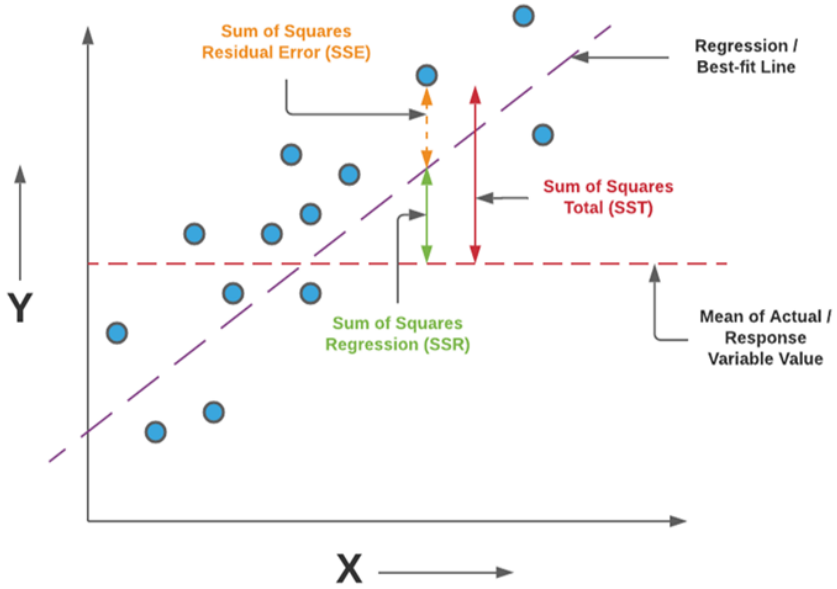
$w_{1_r} = 0.052 \; and \; w_{0_r} = 29.21$

Estimating parameters of Linear Regression.

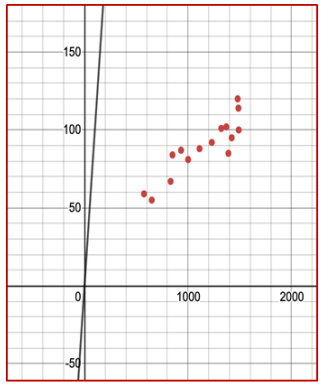
3. An Error Functions

3.1 An Error Function: Residuals.

- In linear regression: a residual is the difference between the actual value and the value estimated by the model **i.e. {residual = actual(y) - estimated($\hat{y}$)}.**
- Residuals and measure of variability:
  - **Sum of Squares Total**
    - $\sum_i^n (y_i - \bar{y})^2$
    - **measures the total variability of the datasets**
  - **Sum of Squares Regression**
    - $\sum_{i=1}^n (\hat{y}_i - \bar{y})^2$
    - **measures the explained variability by your line**
  - **Sum of squares Error**
    - $\sum_{i=1}^n e_i^2$
    - **Here:** $e_i = (y_i - \hat{y})$
    - **measures the unexplained variability by the regression**

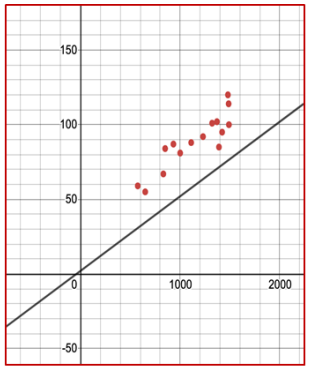


3.2 An Error Function: Demonstrations.



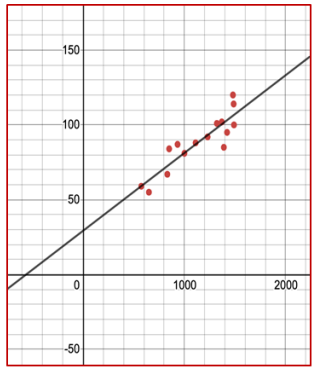
$w_{1_1} = 1 \; and \; w_{0_1} = 0$

X	Actual-Y		Pred-Y	(Pred-Act)^2
575	59		575	266256
650	55		650	354025
832	67		832	585225
850	84		850	586756
933	87		933	715716
1001	81		1001	846400
1111	88		1111	1046529
1230	92		1230	1295044
1321	101		1321	1488400
1370	102		1370	1607824
1390	85		1390	1703025
1422	95		1422	1760929
1480	120		1480	1849600
1487	114		1487	1885129
1490	100		1490	1932100
Squared Errors:				17922958



$w_{1_2} = 0.05 \; and \; w_{0_2} = 2$

Pred-Y	(Pred-Act)^2
30.75	798.0625
34.5	420.25
43.6	547.56
44.5	1560.25
48.65	1470.7225
52.05	838.1025
57.55	927.2025
63.5	812.25
68.05	1085.7025
70.5	992.25
71.5	182.25
73.1	479.61
76	1936
76.35	1417.5225
76.5	552.25
	14019.585



$w_{1_r} = 0.052 \; and \; w_{0_r} = 29.21$

Pred-Y	(Pred-Act)^2
59.11	0.0121
63.01	64.1601
72.474	29.964676
73.41	112.1481
77.726	86.007076
81.262	0.068644
86.982	1.036324
93.17	1.3689
97.902	9.597604
100.45	2.4025
101.49	271.9201
103.154	66.487716
106.17	191.2689
106.534	55.741156
106.69	44.7561
	936.939996

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

$\sum(y_i - \hat{y}_i)^2$ {overall}

Q[1] How do we perform this in computer?

3.3 An Error Function: Definitions.

- aka ~ loss function/Cost Function.
  - **Mean Absolute Error:**
    - We calculate the absolute residual value of all the data points
    - take the average of all these residuals, which gives the magnitude of the residuals
      - $\boldsymbol{MAE = J(\theta, \theta_0) = \left(\frac{1}{N}\right) \sum |y - \hat{y}|}$
      - Does not indicate the under or over performance of the model i.e. contribution of the residuals towards total amount of errors remains proportional.
  - **Mean Square Error:**
    - *squares* the difference before summing them all instead of using the absolute value
      - $\boldsymbol{MSE = J(\theta, \theta_0) = \frac{1}{N} \sum (y - \hat{y})^2}$
      - the error grows **quadratic** in MSE.
- When to use which?
  - **Outliers**
    - Do we include the outliers in our model creation or do we ignore them?
    - dependent on the field of study, the data set on hand and the consequences of having errors in the first place.
    - choice between is MSE and MAE is application-specific and depends on how you want to treat large errors.

Q[2] What kind of plot we get if plotted mse vs. weights?

Linear Regression: Formal Definition.

- Compiling everything till now to formally define Linear Regression:
- In linear regression, the objective (goal) is to fit a hyper plane (a line for 2D data points) by minimizing the sum of mean-squared error for each data point.

Mathematical Interpretation of Linear Regression:

Given a dataset of pairs (X, Y) where:

$X := x_i^d :=$ is a feature vector with property $\in \mathbb{R}^d \; \forall^d \in \{1, \dots, d\}$ [i.e. can have 1 to d many column.]

$Y := y_i :=$ is a scalar with property $\in \mathbb{R} \; \forall_i \in \{1, \dots, n\}$ [i.e. can have n number of rows.]

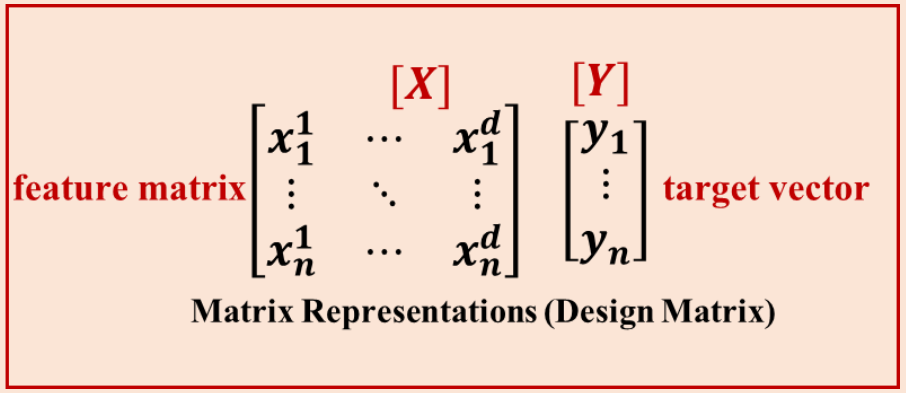
Learn a function:

$$\hat{y} := f_{w_0, w_1}(x) + \epsilon$$

By minimizing a (mean) square error which is:

$$\textbf{square error (pairwise)} := (y - \hat{y})^2$$

$$\textbf{square error (overall)} := \sum (y - \hat{y})^2$$



Matrix Representations (Design Matrix)

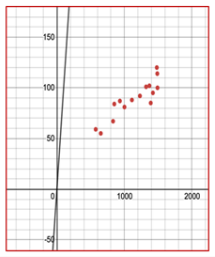
Linear Regression: Story So Far.

Questions:

How do we **optimize** the learning process?

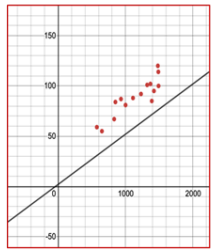
Options-1: Brute Force:

A way to estimate $argmin_{w_0,w_1}\ \mathbb{L}$ is to :
Calculate the loss function for every possible w_0 and w_1.
Then select w_0 and w_1 where the loss function is minimum.



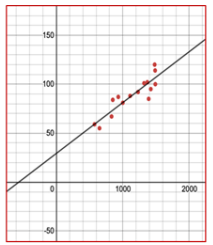
$w_{1_1} = 1\ and\ w_{0_1} = 0$

X	Actual-Y	Pred-Y	(Pred-Act)^2
575	59	575	266256
650	55	650	354025
832	67	832	585225
850	84	850	586756
933	87	933	715716
1001	81	1001	846400
1111	88	1111	1046529
1230	92	1230	1295044
1321	101	1321	1488400
1370	102	1370	1607824
1390	85	1390	1703025
1422	95	1422	1760929
1480	120	1480	1849600
1487	114	1487	1885129
1490	100	1490	1932100
Squared Errors			17922958



$w_{1_2} = 0.05\ and\ w_{0_2} = 2$

Pred-Y	(Pred-Act)^2
30.75	798.0625
34.5	420.25
43.6	547.56
44.5	1560.25
48.65	1470.7225
52.05	838.1025
57.55	927.2025
63.5	812.25
68.05	1085.7025
70.5	992.25
71.5	182.25
73.1	479.61
76	1936
76.35	1417.5225
76.5	552.25
	14019.985



$w_{1_r} = 0.052\ and\ w_{0_r} = 29.21$

Pred-Y	(Pred-Act)^2
59.11	0.0121
63.01	64.1601
72.474	29.964676
73.41	112.1481
77.726	86.007076
81.262	0.068644
86.982	1.036324
93.17	1.3689
97.902	9.597604
100.45	2.4025
101.49	271.9201
103.154	66.487716
106.17	191.2689
106.534	55.741156
106.69	44.7561
	936.939996

Questions:

What kind of plot we get if plotted mse vs. weights?

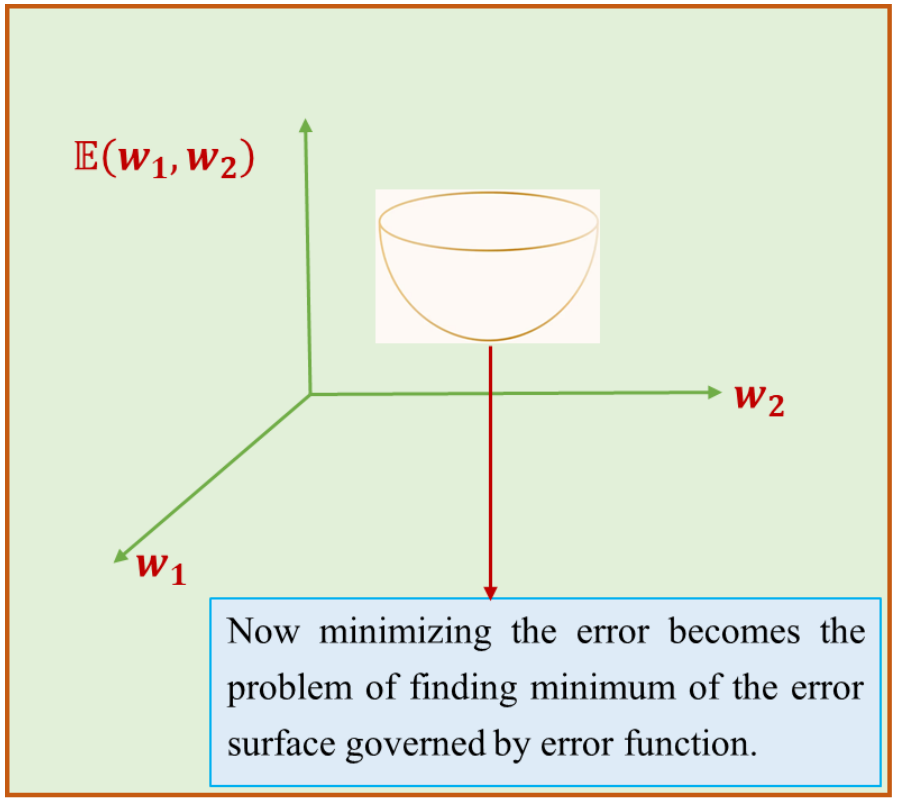
Options-2: Greedy Algorithm.
Gradient Descent.

Learning a Parameters of Linear Regression.

4. An optimization process.

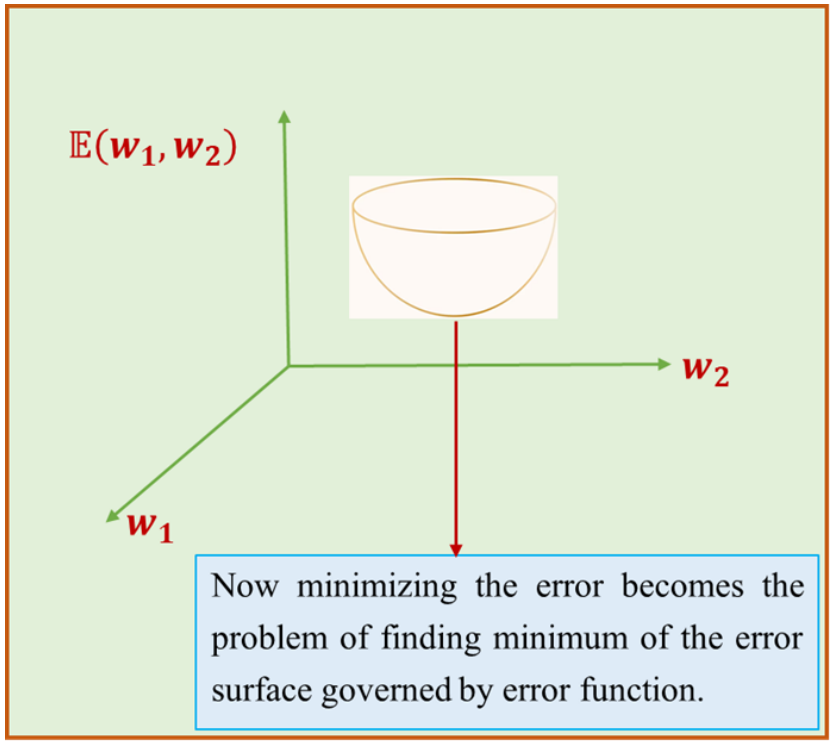
4.1 An optimization Process Error/Loss Surface.

- Question:
  - What kind of plot we get if plotted mse vs. weights?
- **Example:**
  - Consider a linear regression with two inputs $\boldsymbol{X \in \{x_1, x_2\}}$ and assume there is no intercept i.e.
    - $\boldsymbol{\hat{y} = f_{w_1,w_2} = w_1 x_1 + w_2 x_2.}$
  - In this case loss function (square error) will generate a convex error surface with the shape of bowl: →

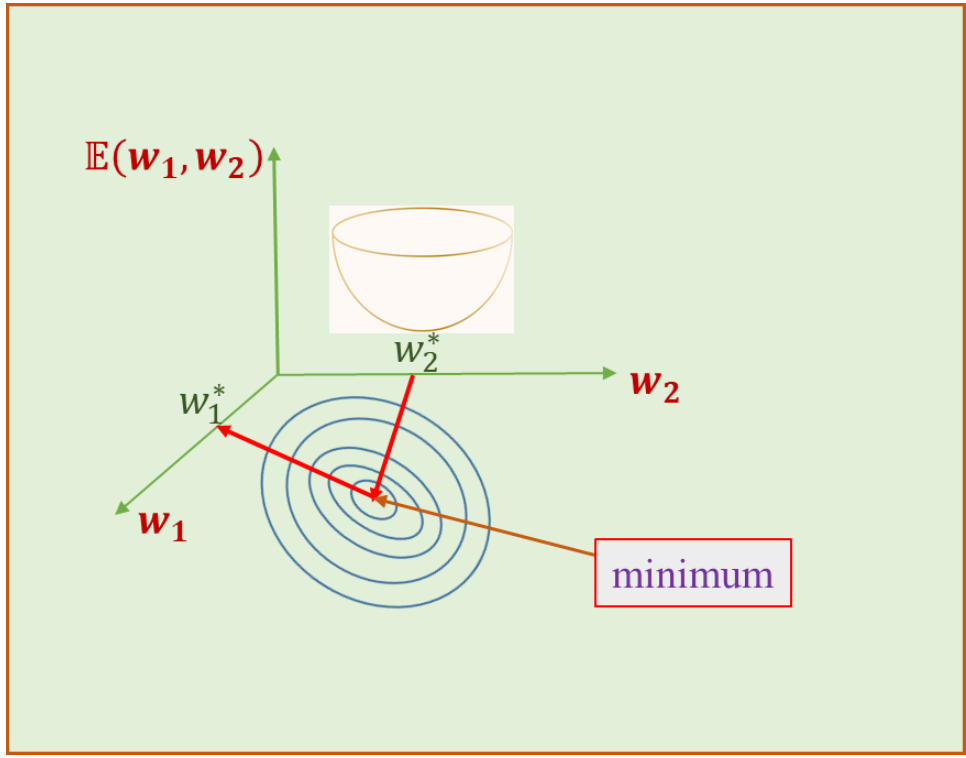


How do we minimize a function?

4.1 An optimization Process Error/Loss Surface.



How do we minimize a function?
Taking a derivative.



Minimizing a **convex function** in Optimization is a Convex Optimization problem.
Such problem can be solved using various iterative algorithms, one such form is:
Gradient Descent!!!!

4.2 What is gradient?

- The **gradient** is a fancy word for derivative, or the rate of change of a function. It's a vector (a direction to move) that
  - Points in the direction of greatest increase of a function.
  - Is zero at a local maximum or local minimum (because there is no single direction of increase).
- The term "gradient" is typically used for functions with several inputs{X} and a single output{Y}.
- **Derivative:**
  - The regular, plain-old derivative gives **us the rate of change of a single variable**, usually x.
    - For example, $\frac{dF}{dx}$ tells us how much the function **F** changes for a change in **x**.
  - But if a function takes multiple variables, such as x and y and z, it will have multiple derivatives:
    - We can represent these multiple rates of change in a vector, with one component for each derivative. Thus, a function that takes 3 variables will have a gradient with 3 components:
      - $F(x, y, z)$ has three variables and three derivatives: $\frac{dF}{dx}$, : $\frac{dF}{dy}$: , : $\frac{dF}{dz}$ {Partial Derivative}
- The gradient of a multi-variable function has a component for each direction.

4.3 Gradient Descent: Idea

- It is an iterative methods used to compute minimum.
- The gradient ∇L at any point is the direction of the steepest increase. The negative gradient is the direction of steepest decrease.
- By following the –ve gradient, we can eventually find the lowest point.
- This method is called Gradient Descent.

4.4 Gradient Descent: Algorithm

- Algorithm:
  - For some cost/loss functions: $\mathbb{E}(w_0, \dots, w_d)$.
  - Start off with some guesses for $w_0, \dots, w_d$
    - It does not really matter what values you start off with, but a common choice is to set them all initially to zero
  - Repeat until Convergence:{

$$w_{new} := w_{old} - \alpha \frac{\partial\, \mathbb{E}(w_0, \dots, w_d)}{\partial w}$$

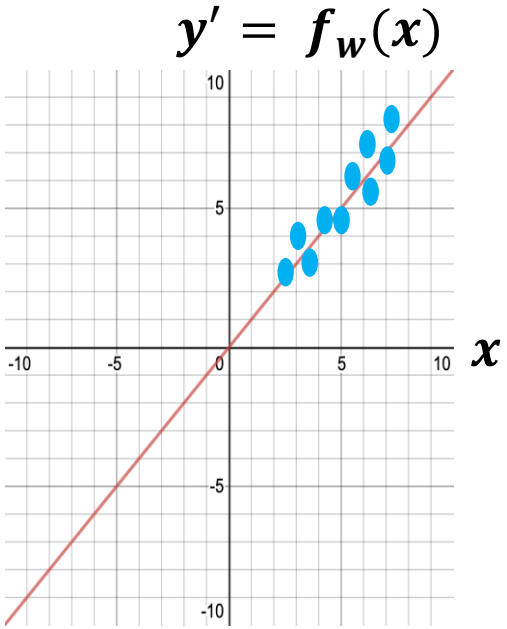
Partial Derivative

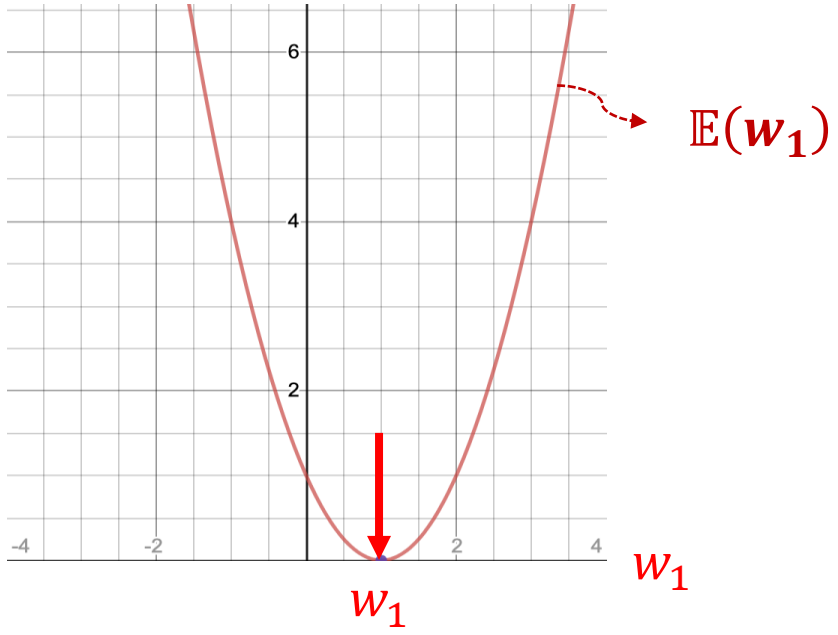
Learing Rate

}

4.5 Gradient Descent - Impact of Partial Derivative.

- Our Objective is to minimize $J(\theta_1)$ for the model represented as $y' = f_w(x) = w_1 x$.



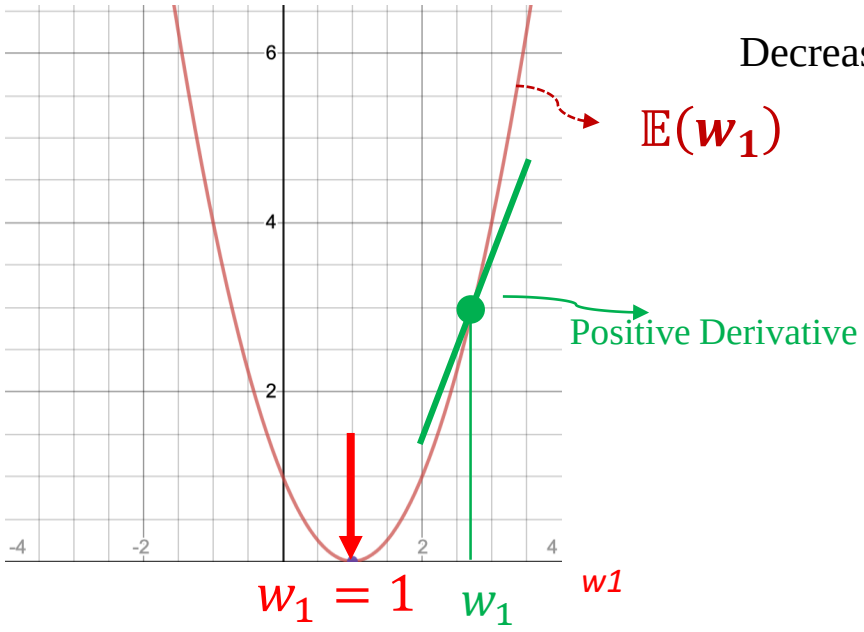


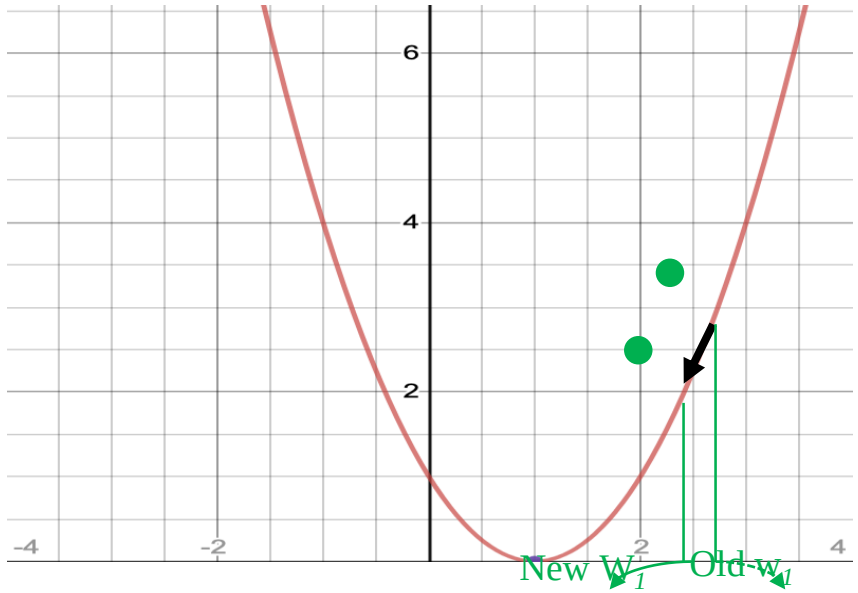
4.5 Gradient Descent-Impact of Partial Derivative.

- Our Objective is to minimize $\mathbb{E}(\boldsymbol{w_1})$ for the model represented as
  - $\hat{y} = \boldsymbol{f_w}(\boldsymbol{x}) = \boldsymbol{W_1 x}.$

$$w_1 = w_1 - \alpha \frac{\boldsymbol{d}\, \boldsymbol{E}(\boldsymbol{w_1})}{\boldsymbol{d}\, w\_1_j}$$
$$= w_1 - \alpha\, (Positive\ Number)$$

Decrease w_1 by a certain value



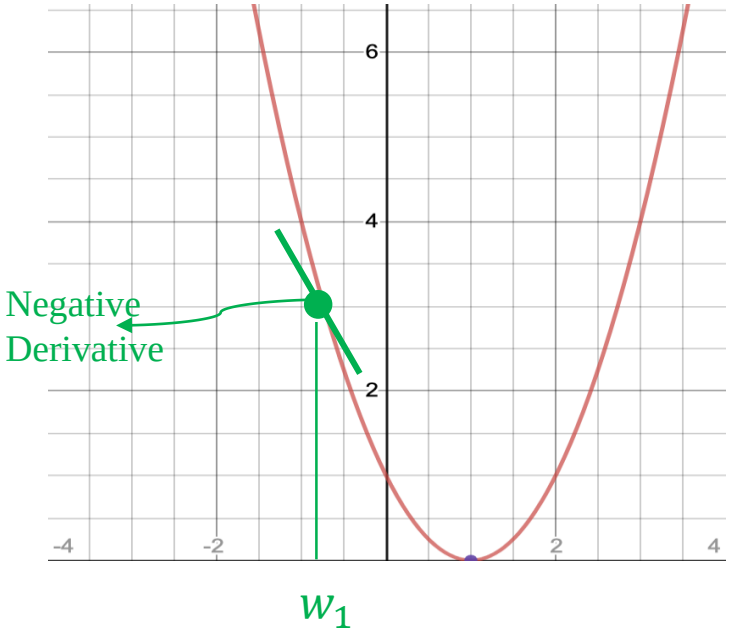


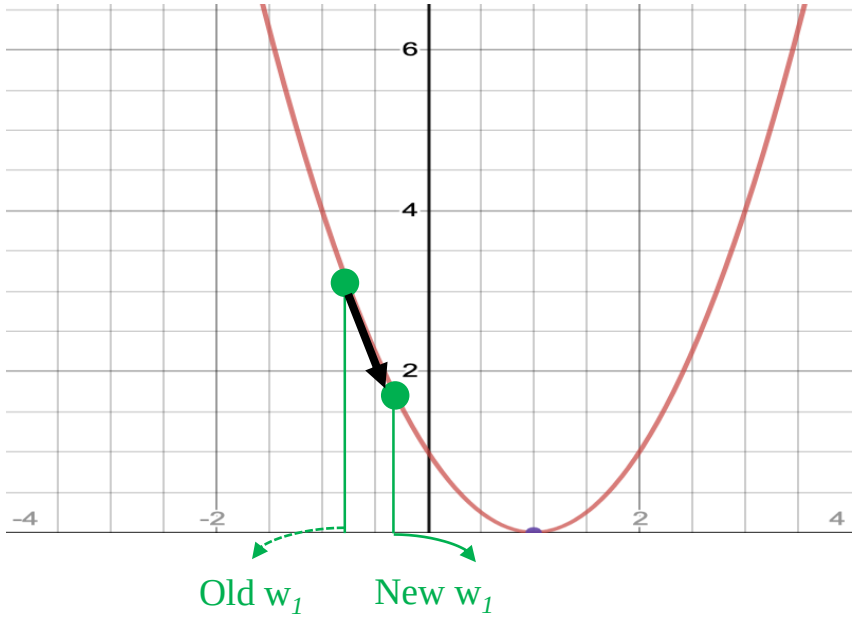
4.5 Gradient Descent-Impact of Partial Derivative.

- Our Objective is to minimize $\mathbb{E}(\boldsymbol{w_1})$ for the model represented as
  - $\hat{y} = \boldsymbol{f_w}(\boldsymbol{x}) = \boldsymbol{W_1 x}.$

$$w_1 = \mathrm{w}_1 - \alpha \frac{\boldsymbol{d}\, \boldsymbol{J}(\boldsymbol{\theta_1})}{\boldsymbol{d}\, \theta_j}$$
$$= \mathrm{w}_1 - \alpha\, (Negative\ Number)$$

Increase w_1 by a certain value



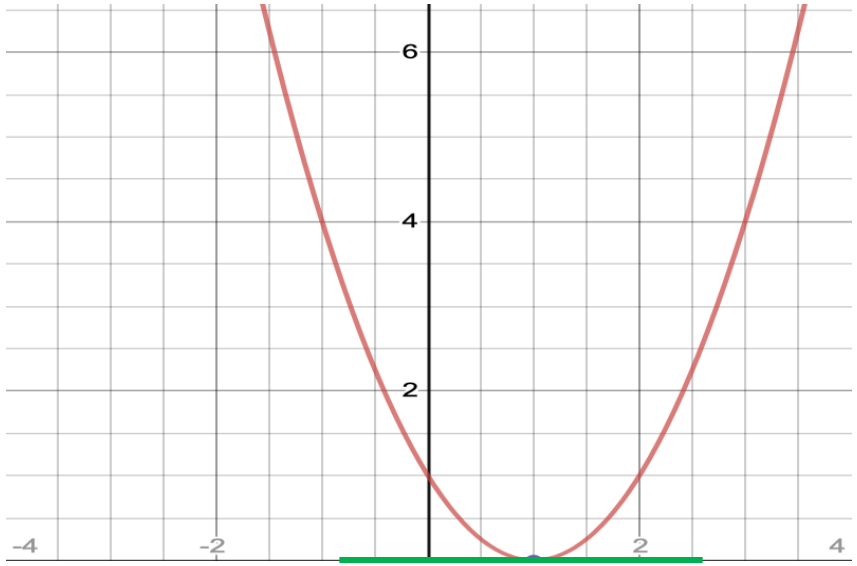


4.5 Gradient Descent-Impact of Partial Derivative.

- Our Objective is to minimize $\mathbb{E}(\boldsymbol{w_1})$ for the model represented as
  - $\hat{y} = \boldsymbol{f_w}(\boldsymbol{x}) = \boldsymbol{W_1 x}.$

$$w_1 = \mathrm{w}_1 - \alpha \frac{\boldsymbol{d}\, \boldsymbol{E}(\boldsymbol{w_1})}{\boldsymbol{d}\, w_j}$$
$$= w_1 - \alpha\, (Zero)$$

w_1 remains same.

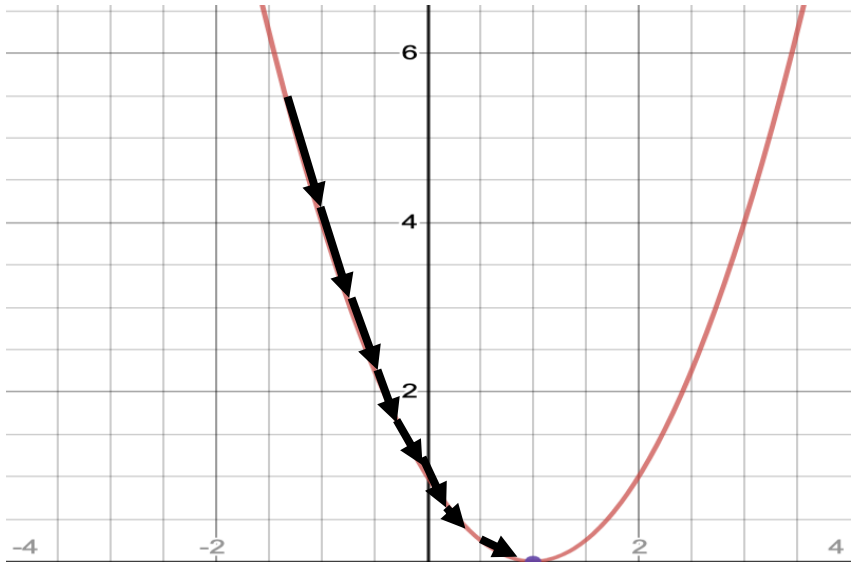


Derivative = 0

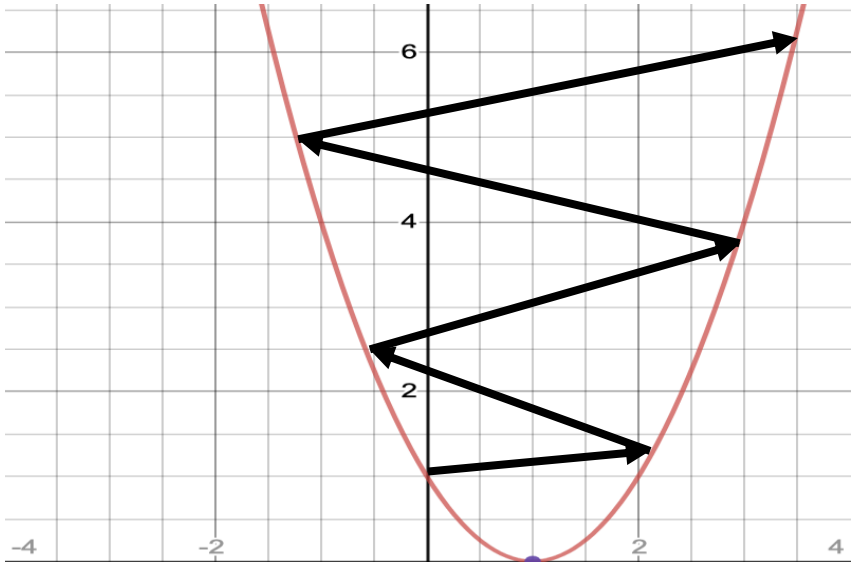
4.6 Gradient Descent-Impact of Learning Rate.

- Our Objective is to minimize $\mathbb{E}(\boldsymbol{w_1})$ for the model represented as
  - $\hat{y} = \boldsymbol{f_w}(\boldsymbol{x}) = \boldsymbol{W_1 x}.$

$$w_1 = w_2 - \alpha \frac{\boldsymbol{d}\, \boldsymbol{E}(\boldsymbol{w_1})}{\boldsymbol{d}w_1}$$
$$= \alpha: Too\ small\ number.$$



$$w_1 = w_1 - \alpha \frac{\boldsymbol{d}\, \boldsymbol{E}(\boldsymbol{\theta_1})}{\boldsymbol{d}\, w_j}$$
$$= \alpha: Too\ Large\ number.$$



4.6 Gradient Descent for Linear Regression.

- Algorithm:
  - For some cost functions: $\boldsymbol{J(w_0, \dots, w_d)}$.
  - Cost\loss function: $\mathbb{L}(w_0, w_1) = \frac{1}{2n}\sum_{i=1}^{n}[h_\Theta(x_i) - y_i]^2$.
  - Start off with some guesses for $w_0, \dots, w_d$
    - It does not really matter what values you start off with, but a common choice is to set them all initially to zero
  - Repeat until Convergence:{

$$\boldsymbol{\Theta_j = \Theta_j - \alpha \frac{\partial \mathbb{L}(w_0, w_1)}{\partial \Theta_j}}$$

$$\frac{\partial \mathbb{L}(\Theta)}{\partial \Theta} = \frac{\partial}{\partial \Theta}\frac{1}{2n}\sum_{i}^{n}[h_\Theta(x_i) - y_i]^2$$

$$= \frac{1}{n}\sum_{i}^{n}(h_\Theta(x_i) - y)\frac{\partial}{\partial \Theta_j}(\Theta x_i - y)$$

$$= \frac{1}{n}[h_\Theta(x_i) - y\,]x_i$$

$$\boldsymbol{\Theta_j = \Theta_j - \frac{\alpha}{n}\sum_{i}^{n}[(h_\Theta(x_i) - y_i)x_i]}$$

  }

Goodness of fit: How good is your Model?

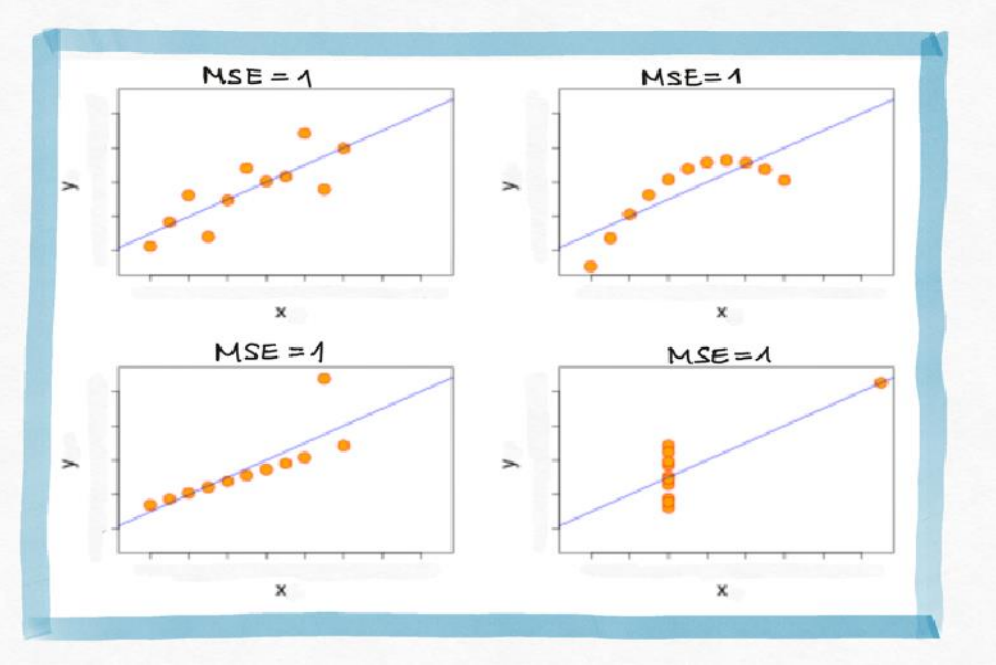
5. Evaluation – Regression Model.

5.1 Model Evaluation

- **R squared or Coefficient of Determination**
  - It can be defined as a Ratio of variation to the Total Variation.
  - The value of R squared lies between 0 to 1, the value closer to 1 the better the model.
    - $R^2 = 1 - \left(\frac{SS_{Res}}{SS_{TOT}}\right) = \sum_i (y_i - \hat{y}_i)^2) / \sum_i (y_i - \bar{y})^2$
- **RMSE**
  - Root of the mean difference of Actual and Predicted values.
  - RMSE penalizes the large errors whereas MSE doesn't.
    - $RMSE = \sqrt{\frac{1}{n}\sum_i (y_i - \hat{y})^2}$

6.2 Model Evaluation: Interpretation.

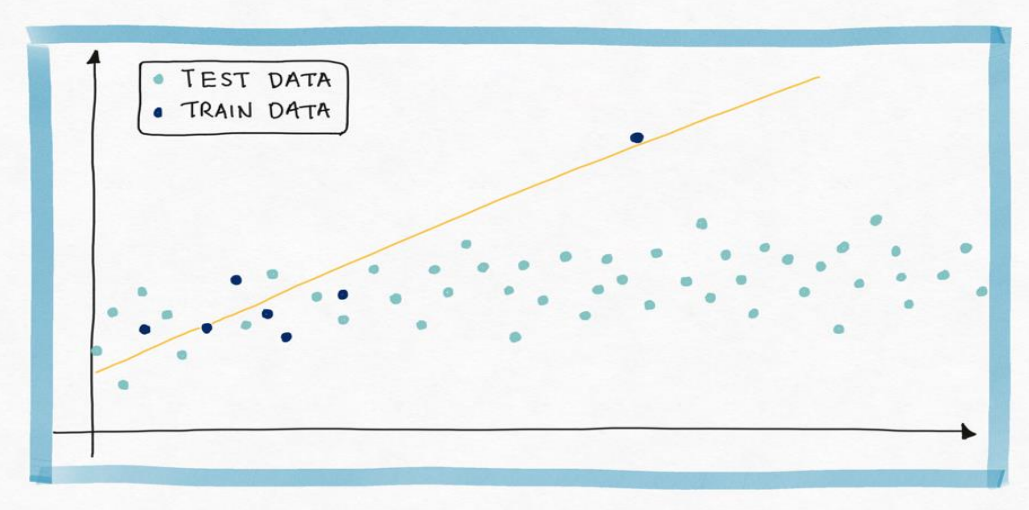
- Just because we found the model that minimizes the squared error it doesn't mean that it's a good model. We investigate the R2 but also:



The MSE is high in all four models but the models are not equal.

6.2 Model Evaluation: Train vs. Test Error.

- How do we achieve the objective of generalizations?
  - We need to evaluate the fitted model on new data, data that the model did not train on, the **test data**.



The **training** MSE here is 2.0 where the **test** MSE is 12.3.

The training data contains a strange point – an outlier – which confuses the model.

Fitting to meaningless patterns in the training is called **overfitting.**

Putting Everything Together!!

7. Vectorization.

7.1 Matrix Representations of Linear Regression

$$Xw = \underset{n\times(d+1)}{\begin{bmatrix} \longleftarrow & x^1 & \longrightarrow \\ \longleftarrow & x^2 & \longrightarrow \\ & \vdots & \\ \longleftarrow & x^n & \longrightarrow \end{bmatrix}} \underset{(d+1)\times 1}{\begin{bmatrix} \uparrow \\ w \\ \downarrow \end{bmatrix}} = \begin{bmatrix} x^1 w \\ x^2 w \\ \vdots \\ x^n w \end{bmatrix} \Rightarrow \begin{bmatrix} \hat{y}^1 \\ \hat{y}^2 \\ \vdots \\ \hat{y}^n \end{bmatrix} = \hat{y}$$

$$Xw = \begin{bmatrix} \sum_{j=0}^{d} x_j^1 w_j \\ \sum_{j=0}^{d} x_j^2 w_j \\ \vdots \\ \sum_{j=0}^{d} x_j^n w_j \end{bmatrix}$$

7.1 Matrix Representations of: SSE Loss

- Representing (mean) SSE in matrix notation:
- $\mathbb{E}(w) = \|y - \hat{y}\|^2$

$$y - \hat{y} = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^n \end{bmatrix} - \begin{bmatrix} \hat{y}^1 \\ \hat{y}^2 \\ \vdots \\ \hat{y}^n \end{bmatrix} = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^n \end{bmatrix} - \begin{bmatrix} x^1 w \\ x^2 w \\ \vdots \\ x^n w \end{bmatrix} = \underbrace{\begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^n \end{bmatrix}}_{n \times 1} - \underbrace{\begin{bmatrix} \longleftarrow & x^1 & \longrightarrow \\ \longleftarrow & x^2 & \longrightarrow \\ & \vdots & \\ \longleftarrow & x^n & \longrightarrow \end{bmatrix}}_{n \times (d+1)} \underbrace{\begin{bmatrix} \uparrow \\ w \\ \downarrow \end{bmatrix}}_{(d+1) \times 1}$$

Next Week:

- Answer to Question:
  - **What if target variables are discrete (Linear Methods for Classifications)?**
  - **What are the methods we can try to achieve the generalizations?**
  - **What is Training and Test Error?**
  - **What does Overfitting means?**
    - **We will answer all the question by defining the concept of bias-variance tradeoff.**

Thank You any Question!!!