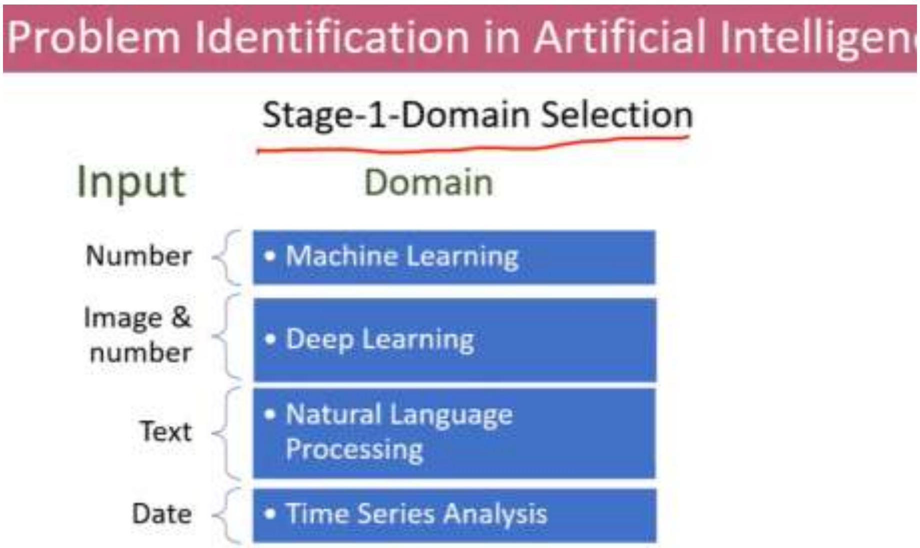
MACHINE LEARNING

1) Problem identification

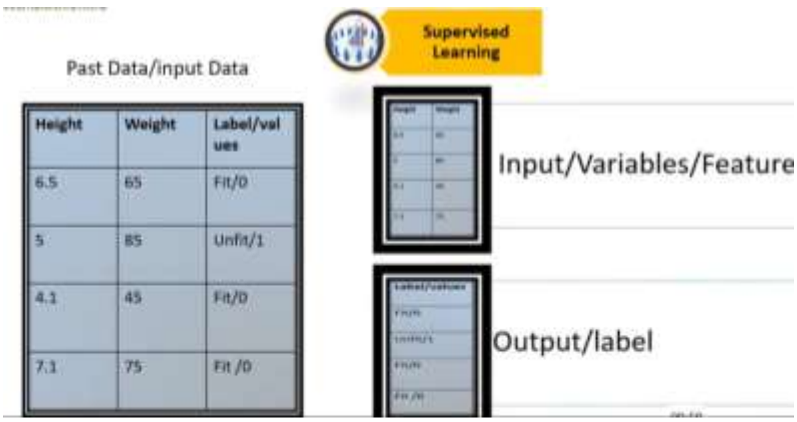


2) Learning Session:

- Supervised Learning
  - Requirement is clear, Dataset(input/output) clear.
- Unsupervised Learning
  - Only input variable
  - Clustering – it is input for target marketing (Specific group of people target)
- Semi –Supervise Learning

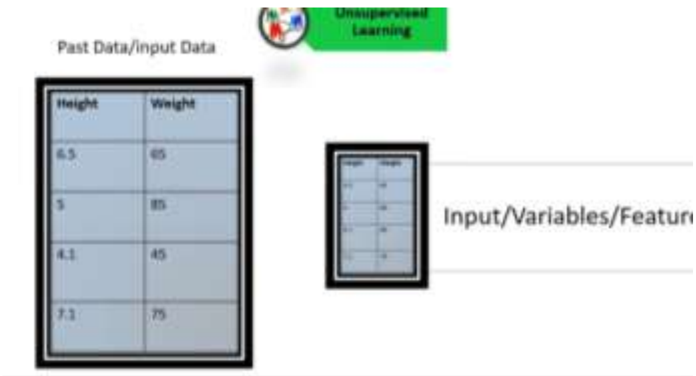
- Combination supervised and semi-supervised learning
- Requirement have but output not clear

3) Supervised Learning:

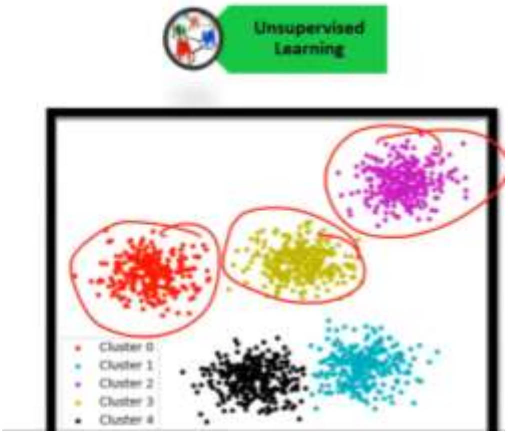


Input -> Independent Variable
Output-> Response Variable

4) Unsupervised Learning:

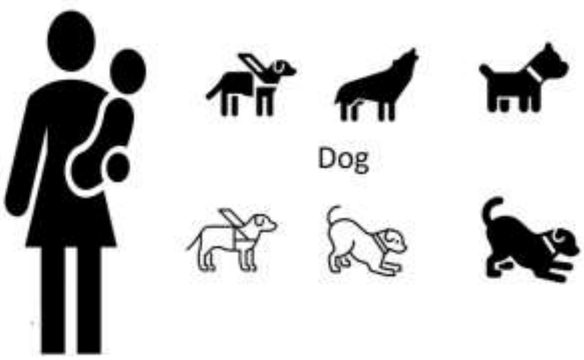


It can be grouped into clusters:

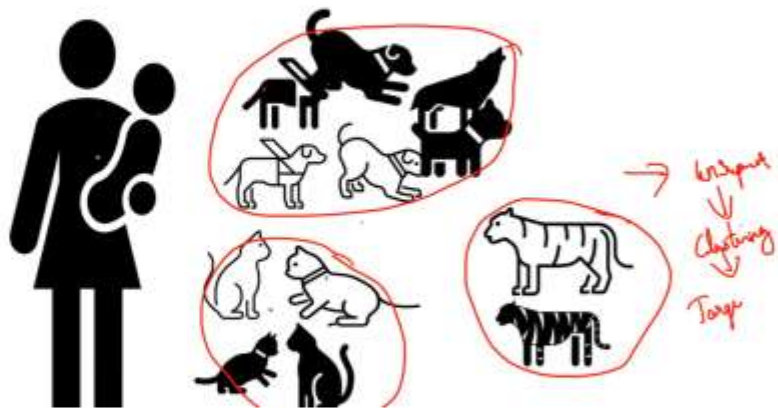


5) Difference between unsupervised and supervised learning

Supervised Learning: we can guess output from particular input

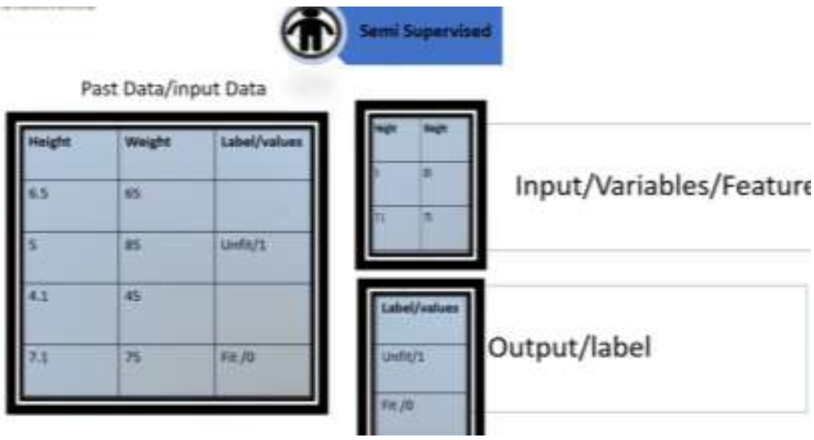


Unsupervised Learning: we have many input to filter and targeting by grouping to predict output.



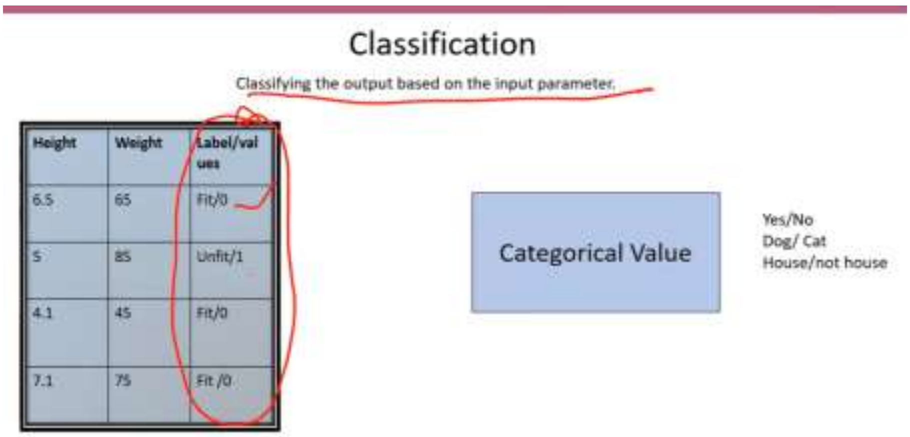
6) Semi supervised Learning:

-Requirement is clear. But output is not clear

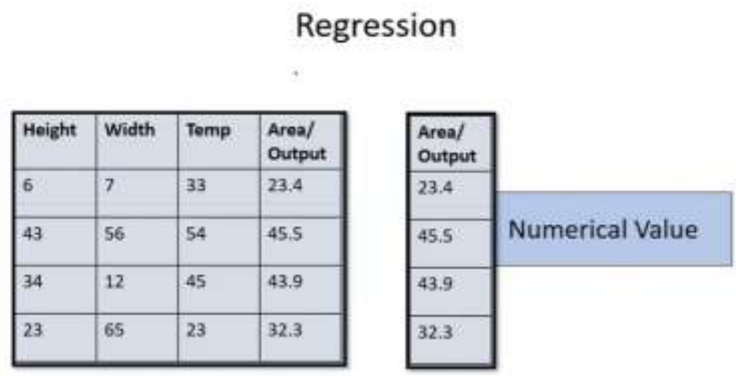


7) Supervised learning under 3 stage after identify by Regression or Classification:

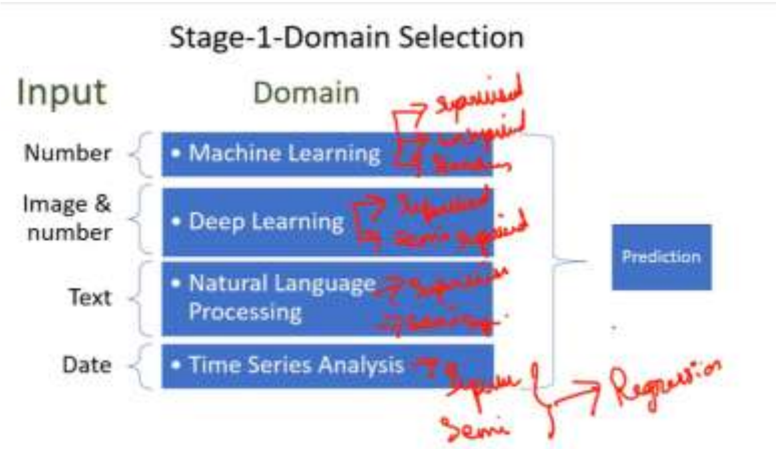
Classification: Output label either this/that



Regression: Output label should be Continues value/numerical value



8) Stage Domain Classification:



Machine learning ->

Supervised, Semi-supervised, Unsupervised learning all support

Deep learning ->

Supervised, Semi-supervised only support

Natural Language Processing ->

Supervised, Semi-supervised only support

Time –series Analysis ->

Supervised, Semi-supervised only under Regression support.

9) Problem Identification Real Time Scenario 1:

Hope Artificial Intelligence

The XYZ bank facing the issues of fraudulent, before making false transaction system should prevent from the fraudulent. As AI engineer how will you solve the problem?

A) Supervised-Regression
B) Supervised -Classification
C) Unsupervised- Clustering
D) Semi-Supervised

Select the best answer and justify why you have chosen the same.
In justification you can discuss the dataset, call to action. e.t.c.,

Stage 1: Guess -> transaction under number, so Machine Learning/Deep Learning (if required)

Choose -> Machine Learning.

Stage 2: Learning from this input is clear that is amount transaction (only one)

Choose-> Supervised

Stage 3: Output predict fraud/not fraud

Choose-> Classification

Dataset: {Input1, Input2….} -> Output: Fraud/Non-fraud

Answer: Supervised Classification

10) Problem Identification Real Time Scenario 2:

The XYZ is an E-commerce company. Daily they receive more than 1000 of mails regarding shipment, product delivery, Quality of product and etc. Issue is every customer mail to one mail id but company have different domains to solve these issues for the customer. As an AI Engineer, you are asked to do redirect the mail from customers complaints mail id to respective department mail id, so that respective department will resolve the issues.

Identify the 3 Stages of Problem Domain. DataSet

Data set ->

Input	Output
Mail	Shipment
Mail	Delivery
Mail	Quality
Mail	Issue

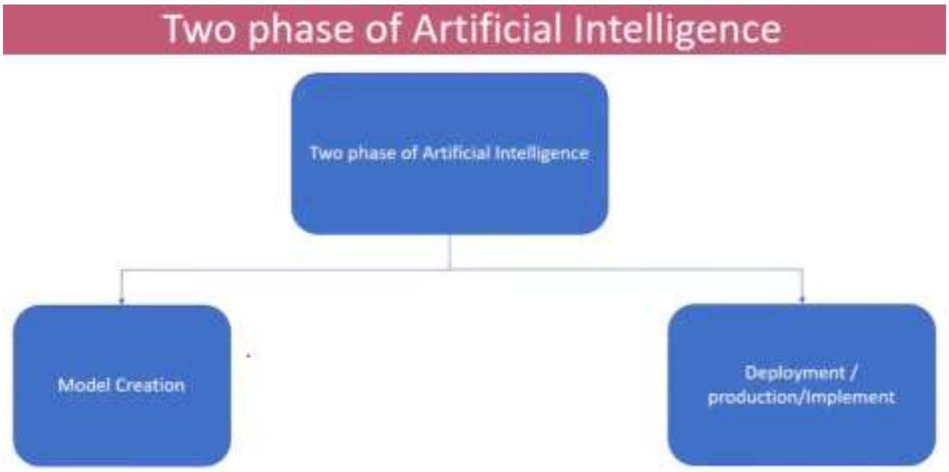
Stage 1: All data in "Text" format -> NLP

Stage 2: Requirement of input and output is clear -> Supervised

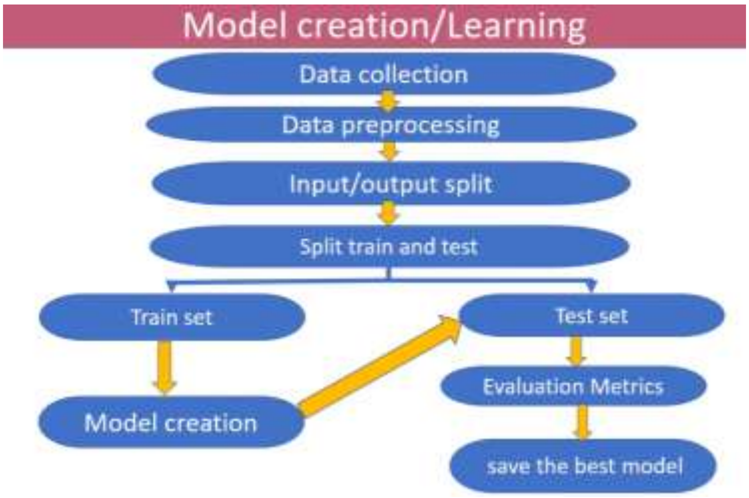
Stage 3: Either this or that -> Classification

Answer: NLP -> Supervised -> Classification

11) Phase of AI:

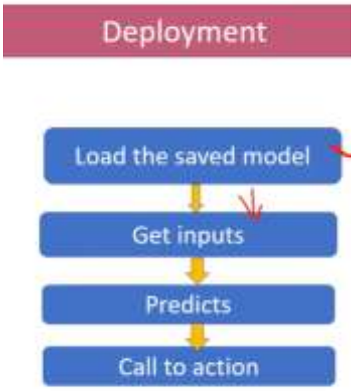


MODEL CREATION:

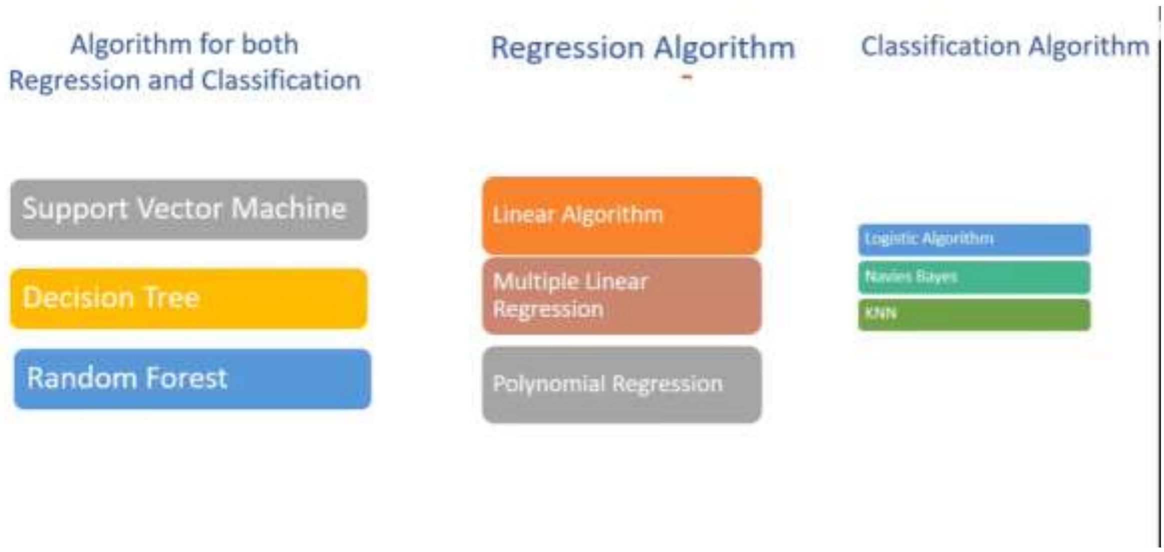


Train set-> Revision

DEPLOYMENT:



12) Machine Learning Algorithms:



13) Simple Linear Algorithm:

It has one input(X/Independent) and one output (y/Dependent Variable/Response variable).

It uses the straight line equation y= mX+c

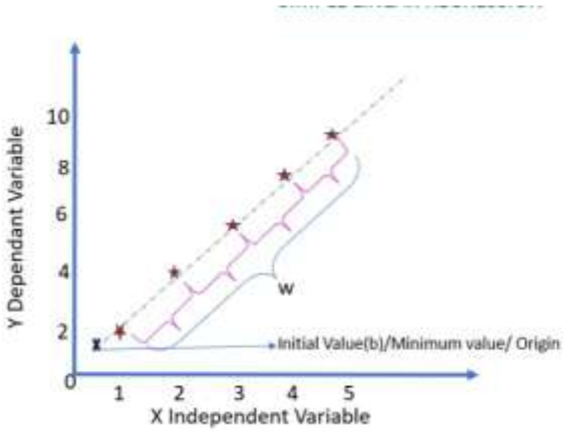
Where,

y= output(which forms straight line by adding all data points)

m= slope= dy/dx= weight (which says about constant distance between two data points)

X= Input(if input changes, then output will also change accordingly)

c= bias= intercept= starting of the straight line= initial value= minimum value.



Dataset

X	Y
1	2
2	4
3	6
4	8
5	10

Y = w X + b

W= Slope= $\frac{(\sum y)(\sum x^2) - (\sum x)(\sum xy)}{n(\sum x^2) - (\sum x)^2}$

B= Bias=Initial Value=Minimum Value= $\frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2}$

Stage 1: Machine Learning
Stage 2: Supervised Learning
Stage 3: Regression

Y =wx+b / mx+c

W -> formula already defined in "Python"

y= wx+b → SLR → F
↓
dataset → w, b
↓
$\boxed{y = 2x+0}$
↓
Model

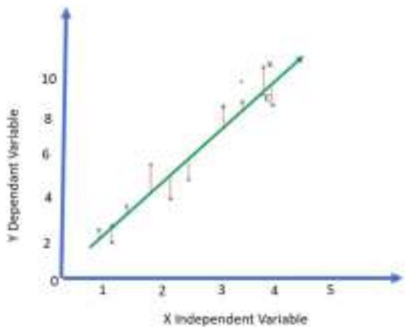
Y=2x+0 -> Model

14) Evaluation Metris:

Validating the Algorithm output predicted value that how accurate our result has – that is finding Regression

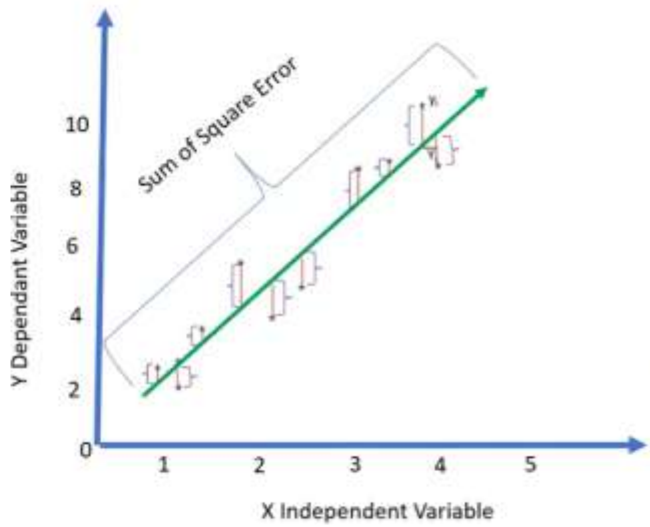
Real-time comparison:
Total Score -> 500
Boy 1 scored -> 498 (2 Error), Boy 2 Scored -> 490 (10 error)

Validating parameter: 1 .Sum of Square Error(SSE) or Residual Sum of Square(RSS)



Error:

Input	Actual Output	Predicted Output	Error=(Actual -Predicted)^2
1	3.8	3.5	0.09
3	4.5	4.7	0.02
4	5.6	5.3	0.09
5	4.6	1.4	10.24
6	2.3	3.4	1.44
9	7.6	7.1	0.25
10	3.4	2.3	0.87



Formula:

$$Sum\ of\ Square\ Error(SSE) = \sum_{i=0}^{n}(y_i - \bar{y_i})^2$$

Where, i = Observation point

n= number of Observation point

$\bar{y}_i$= Predicted Value

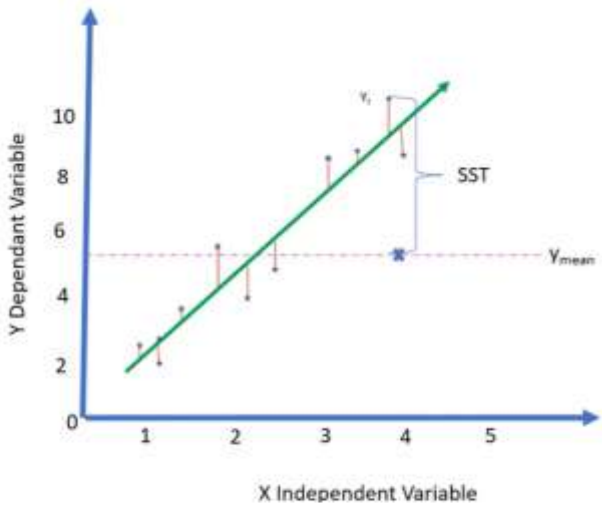
y_i=Actual Value

Take away:

If,

Higher the SSE, then predicted value is poor
Smaller the SSE, then predicted value is good

Total Sum of square -> Average finding (SUMMATION)



Formula:

$$Sum\ Square\ Total(SST) = \sum_{i=0}^{n}(y_i - ymean)^2$$

SST= SSR+SSE

Where, i = Observation point

n= number of Observation point

y_i= Actual Value

y_{mean}= Mean of Dependant Variable(Response variable)

Validating parameter: 4. R Squared(R^2)

$$R^2 = \frac{SSR}{SST} = \frac{\sum_{i=0}^{n}(\bar{y}_i - ymean)^2}{\sum_{i=0}^{n}(y_i - ymean)^2}$$

The above all metrics is based on comparison with 2 value otherwise we cannot judge whether given output is how much accurate. So that reason finding R SQUARED.

$$R^2 = \frac{SSR}{SST} = \frac{\sum_{i=0}^{n}(\bar{y}_i - ymean)^2}{\sum_{i=0}^{n}(y_i - ymean)^2}$$

Where, i = Observation point

n= number of Observation point

$\bar{y}_i$= Actual Value

y_i=Predicted Value

y_{mean}= Mean of Dependant Variable(Response variable)

Purpose of R^2 :

To know , how well the model is fitted.

How R^2 differs from other parameters like SSE,SSR and SST ?

SSE, SSR and SST range varies with dataset to dataset.

But,

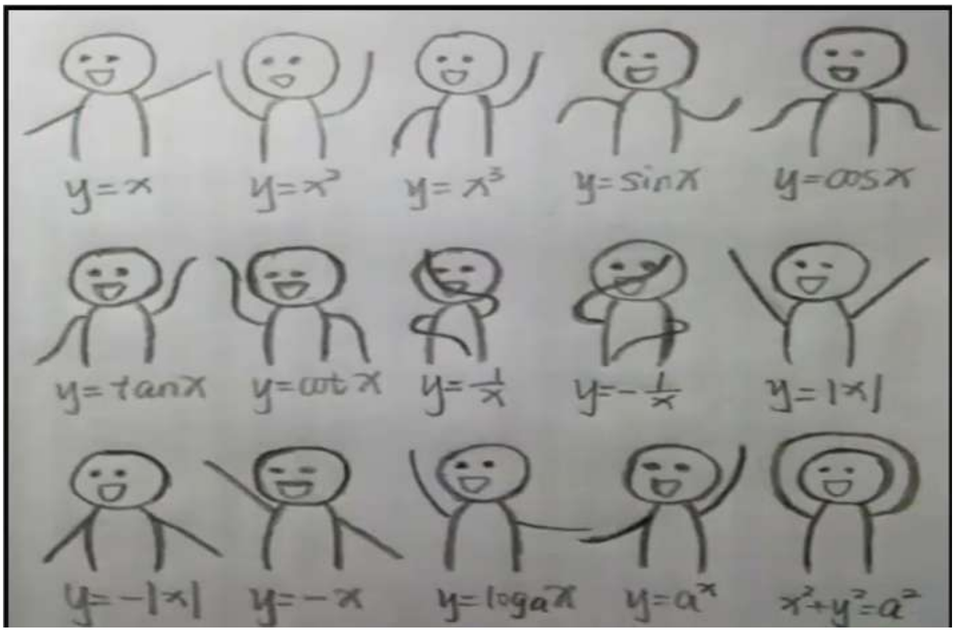
R^2 exists between 0 and 1

If,

R^2 = nearly to 1, then built model has better performance.

R^2 = nearly to 0, then built model has poor performance.

The only drawback of R^2 is that if new predictors (X) are added to our model, R^2 only increases or remains constant but it never decreases. We can not judge that by increasing complexity of our model, are we making it more accurate



First Row:

- y = x: The stick figure's arm is diagonally raised, matching the straight, upward-sloping line of the function y = x.
- y = x^2: Both arms are raised in a U-shape, resembling the parabola opening upwards.
- y = x^3: One arm curves steeply upward and the other downward, reflecting the cubic function's S-shaped curve.
- y = sin(x): The arms are wavy, imitating the periodic, oscillating pattern of the sine function.
- y = cos(x): The arms are outstretched horizontally, similar to the cosine curve's starting point at its maximum.

Second Row:

- y = tan(x): One arm is thrown sharply upward, the other downward, representing the tangent function's steep slopes and asymptotes.
- y = cot(x): Arms are curved in the opposite direction to tan(x), reflecting the cotangent's behavior.
- y = 1/x: One arm is raised, the other lowered, mimicking the hyperbola's two branches.
- y = -1/x: The arm positions are swapped compared to y = 1/x, indicating the negative reciprocal function.
- y = |x|: Both arms are raised in a V-shape, like the absolute value function's graph.

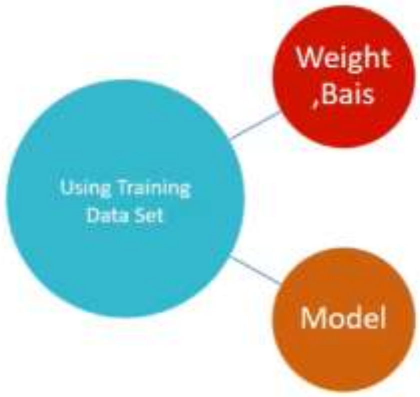
Third Row:

- y = -|x|: Both arms are downward in an inverted V, reflecting the upside-down absolute value graph.
- y = -x: One arm is diagonally downward, matching the negative slope of the line y = -x.
- y = $\log_a x$: One arm is extended to the side and curves upward, showing the slow growth of the logarithmic function.

- $y = a^x$: One arm is low, the other curves steeply upward, resembling the rapid increase of the exponential function.
- $x^2 + y^2 = a^2$: Both arms are curved above the head in a circle, representing the equation of a circle.

15) Testing and Train Set

Using Training data set 80, I will test the model of test set 20, Y= 0,3x1+0.4x2+0 (MODEL)



$Y= 0.3x_1+0.4x_2+0$

Height x_1	Weight x_2	Label/values y
6.5	65	Fit/0
5	85	Unfit/1
4.1	45	Fit/0
7.1	75	Fit /0

$Y= 0.3x_1+0.4x_2+0$ Found using Training Set

$y = (0 \cdot 3 * 6.5) + (0 \cdot 4 * 65)+0 = 27.95$
$y = (0 \cdot 3 * 5) + (0.4 * 85) + 0 = 35.5$

```
If(y>30):
        print("Unfit")
Else
        print("fit")
```