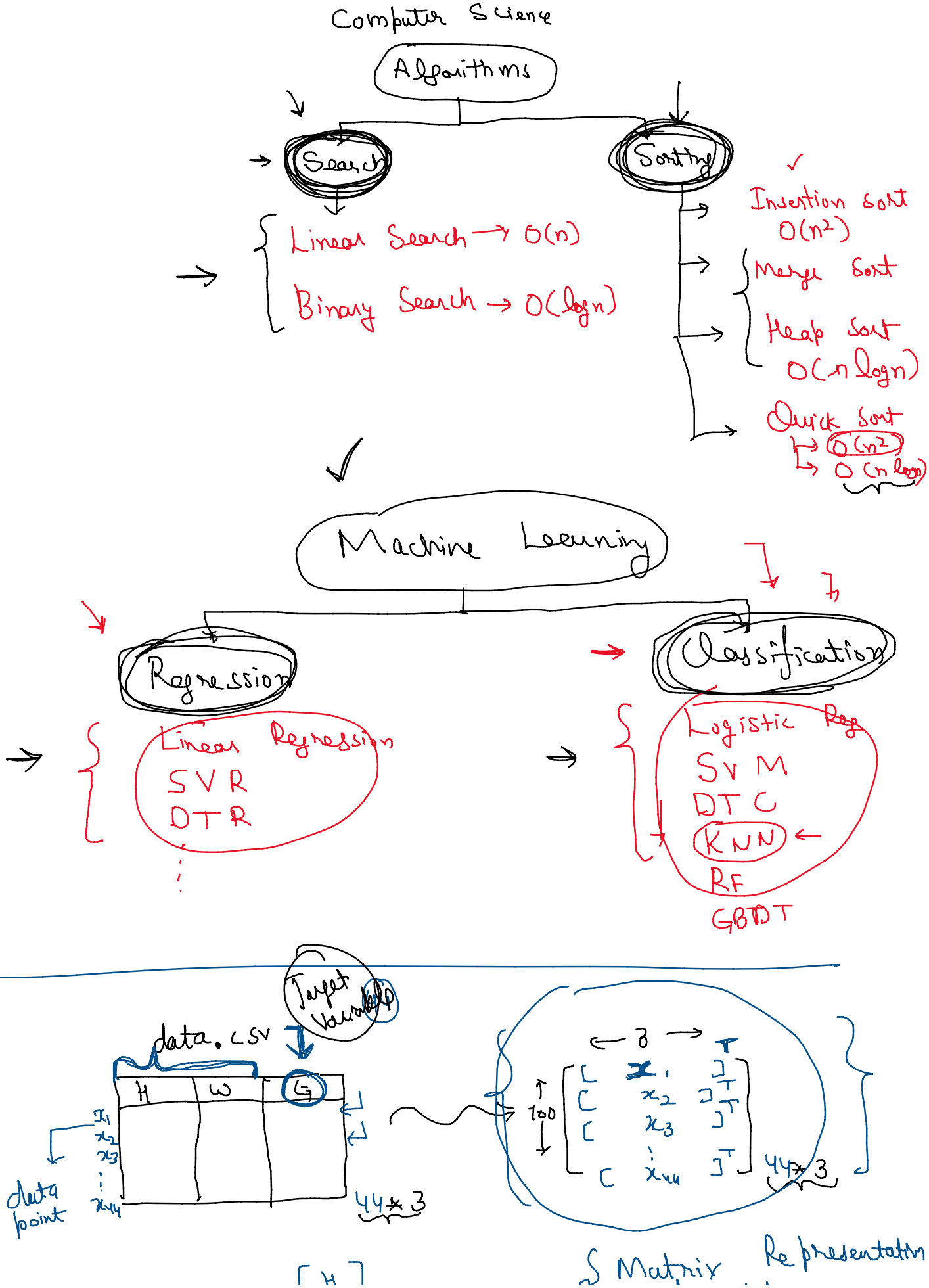
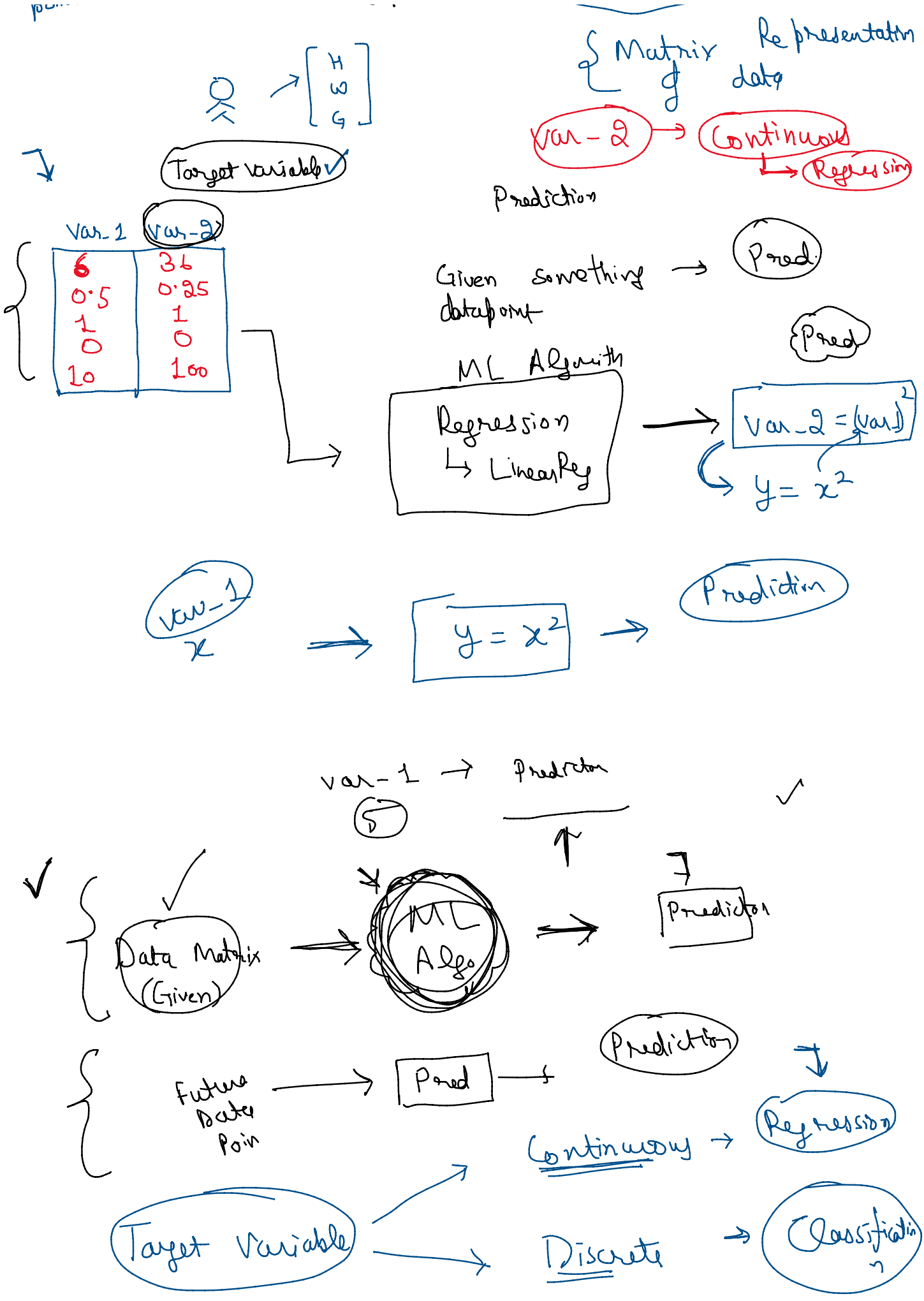
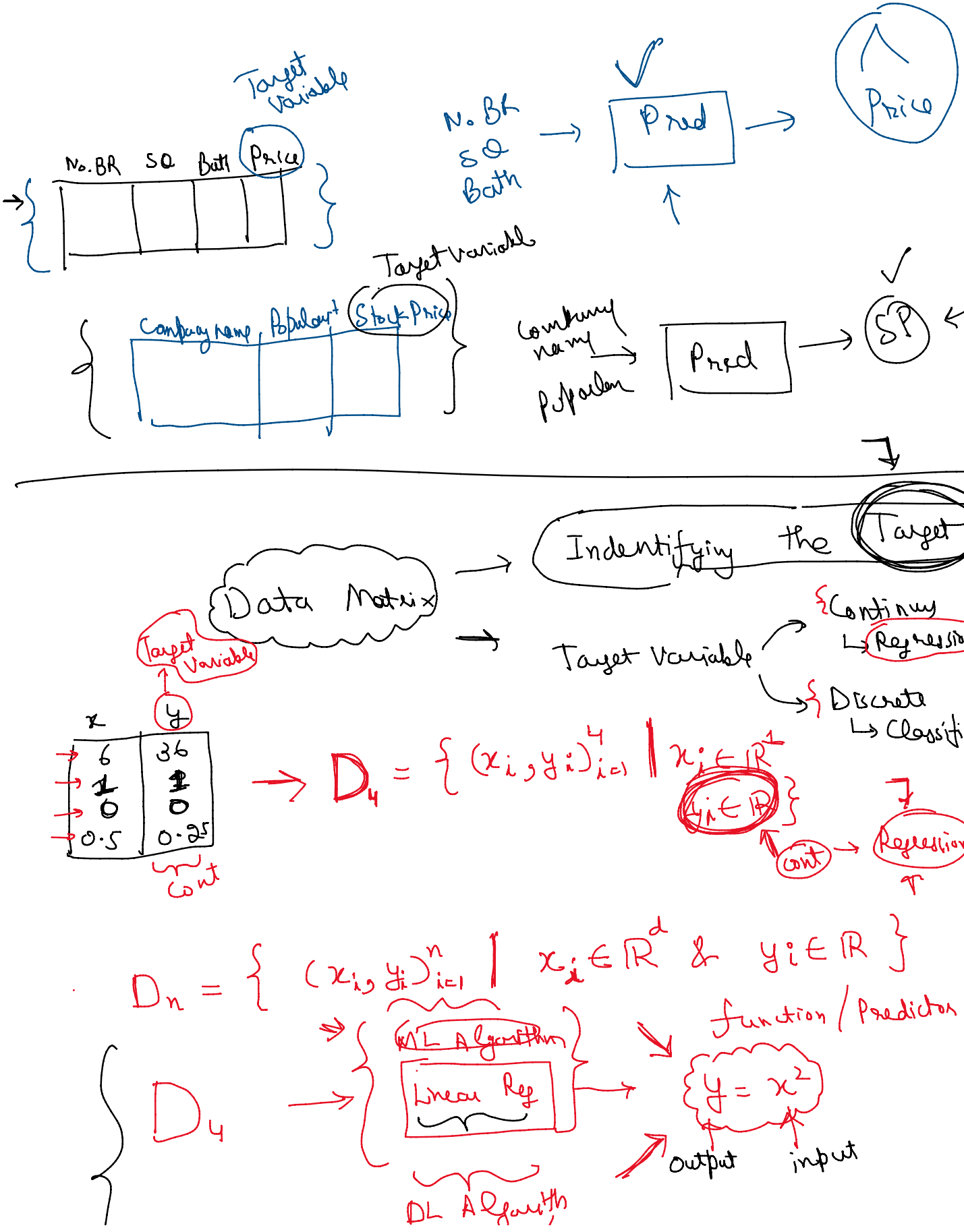
Machine Learning

22 July 2020 06:03 PM




Matrix
Representation
data
H
W
G
var-2
Continuous
Regression
Target variable
Prediction
var_1
var-2
6
0.5
1
0
10
36
0.25
1
0
100
Given something
datapoint
Pred
ML Algorithm
Regression
LinearReg
var_2 = (var_1)^2
y = x^2
var-1
x
y = x^2
Prediction
var-1 → Prediction
5
Data Matrix
(Given)
ML
Algo
Prediction
Future
Data
Poin
Pred
Prediction
Regression
Continuous →
Target Variable
Discrete
Classification



$$D_4 = \{ (x_i, y_i)_{i=1}^{4} \mid x_i \in \mathbb{R}^1 \ \& \ y_i \in \mathbb{R} \}$$

$$D_n = \{ (x_i, y_i)_{i=1}^{n} \mid x_i \in \mathbb{R}^d \ \& \ y_i \in \mathbb{R} \}$$

DL Algorithm

$x = 10 \rightarrow y = x^2 \rightarrow y = 100$ (output)

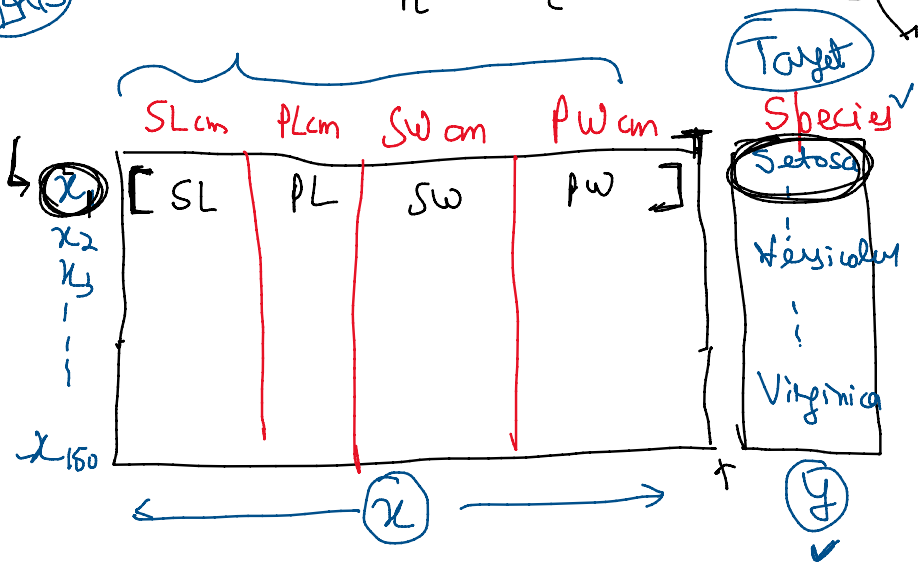
Ques - 1 :

→ Given a dataset (input)

$$D_n = \{ (x_i, y_i)_{i=1}^{n} \mid x_i \in \mathbb{R}^q, \; y_i \in \{\text{Setosa, Versicolor, Virginica}\} \}$$

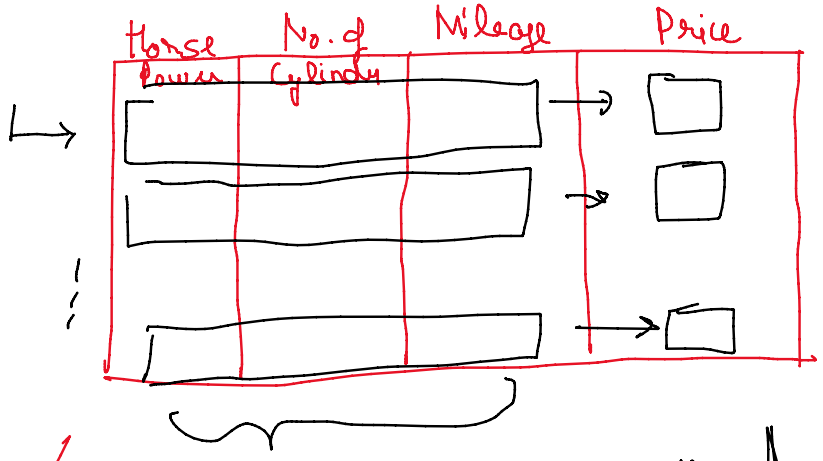
(y_i : output)

Iris



y_i → Target Variable
↳ Discrete Value
↳ Classification
↳ Logistic Reg

Ques - 2 ↓ Given data ↓



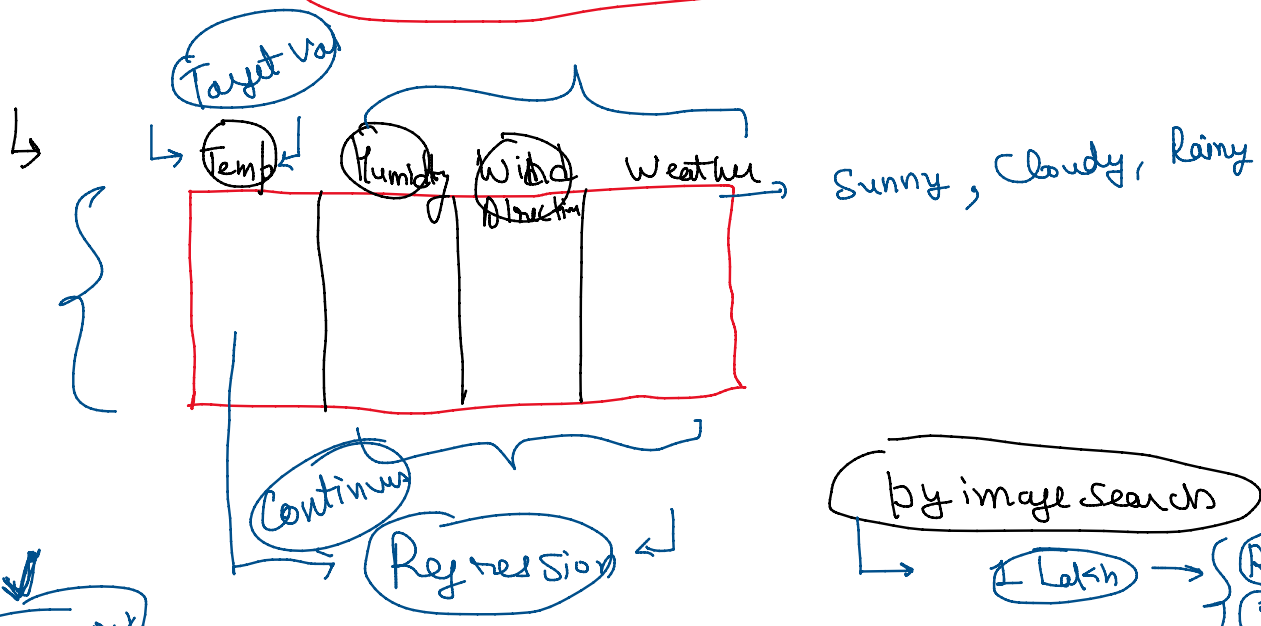
→ D

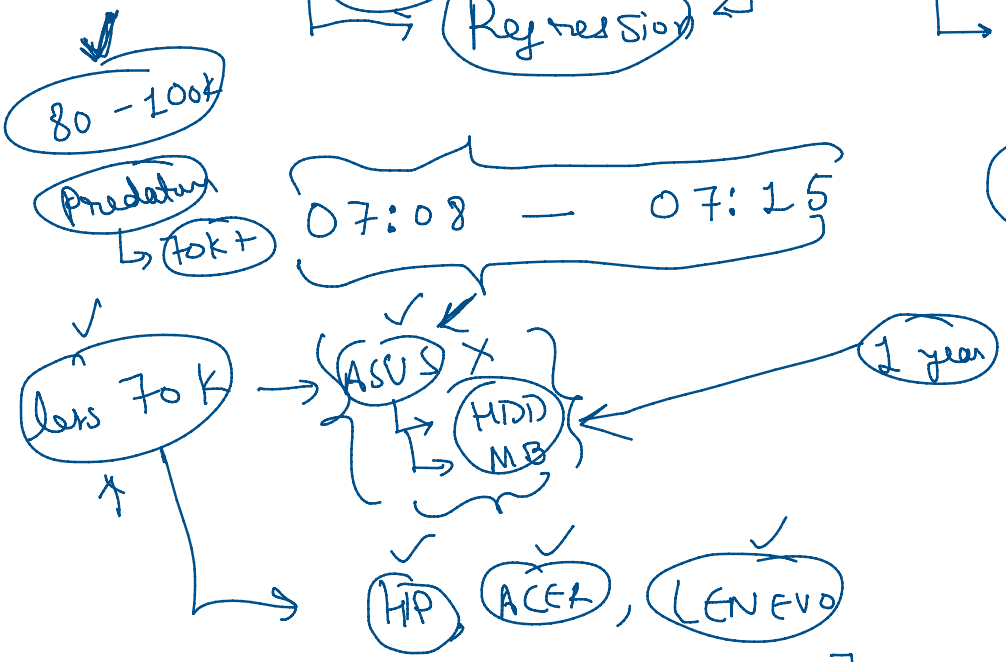
↳ $D_n = \{ (x_i, y_i)_{i=1}^{n} \mid x_i \in \mathbb{R}^3, \; y_i \in \mathbb{R} \}$ ✓

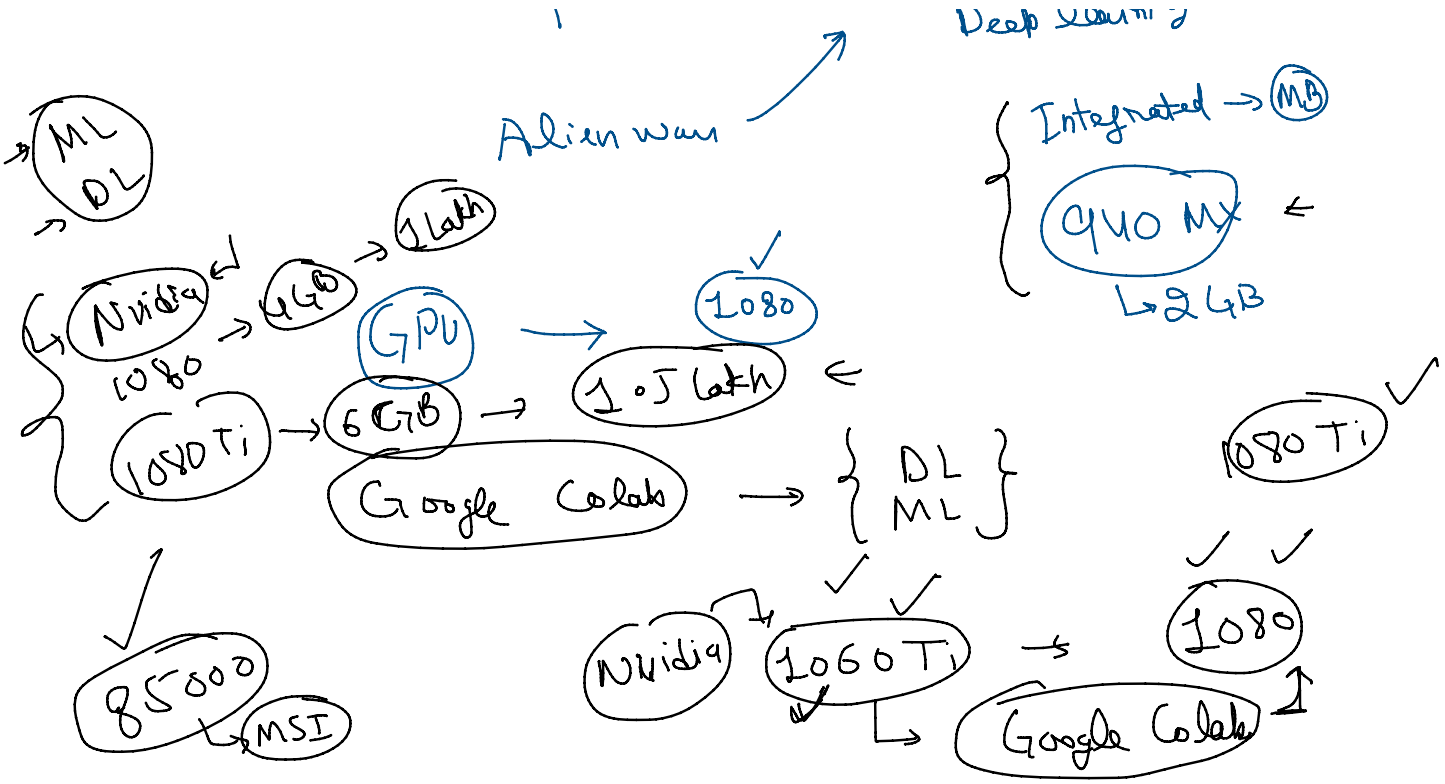
(input; y_i : Target variable)

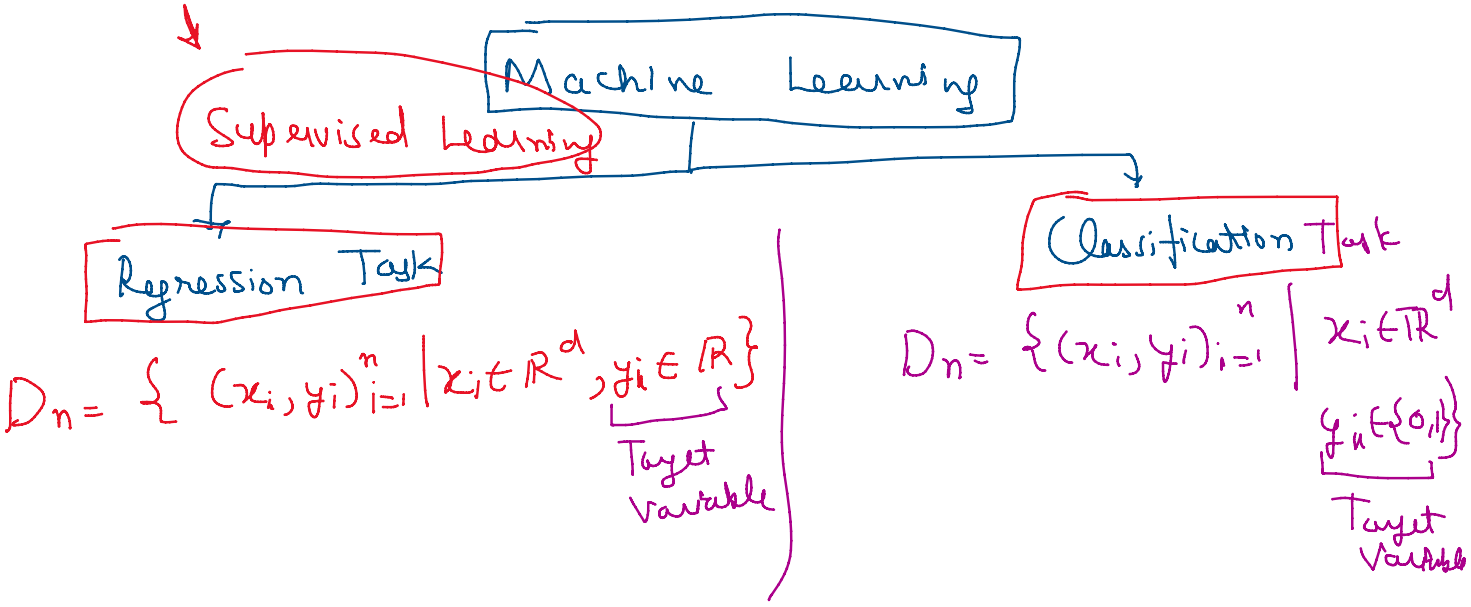
Target variable is continuum
↳ Regression

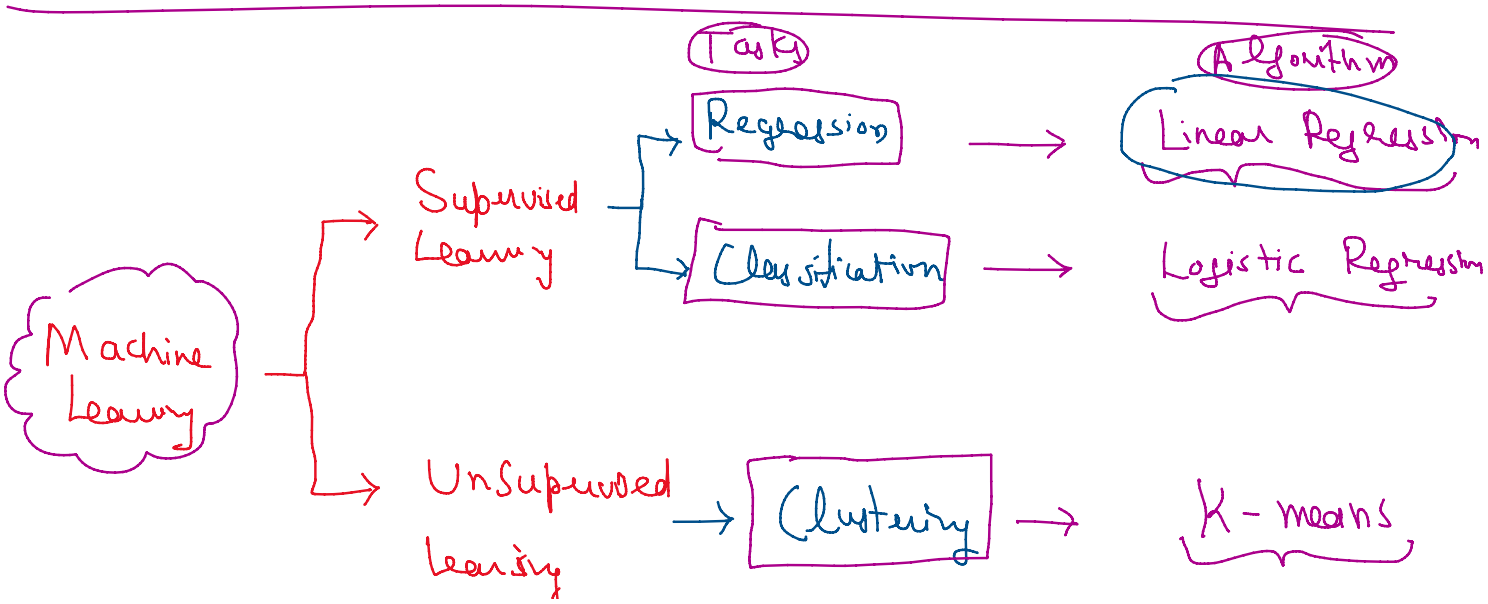
Linear Regression






ML
DL
Alienware
Integrated → MB
940 MX
↳ 2 GB
Nvidia
1080
4 GB
1 Lakh
1080 Ti
GPU
6 GB
1080
10 Lakh
Google Colab
DL
ML
85000
MSI
Nvidia
1060 Ti
1080 Ti
1080
Google Colab


Machine Learning
Supervised Learning
Regression Task
Classification Task
$D_n = \{(x_i, y_i)_{i=1}^n \mid x_i \in \mathbb{R}^d, y_i \in \mathbb{R}\}$
Target variable
$D_n = \{(x_i, y_i)_{i=1}^n \mid x_i \in \mathbb{R}^d$
$y_i \in \{0,1\}$
Target variable

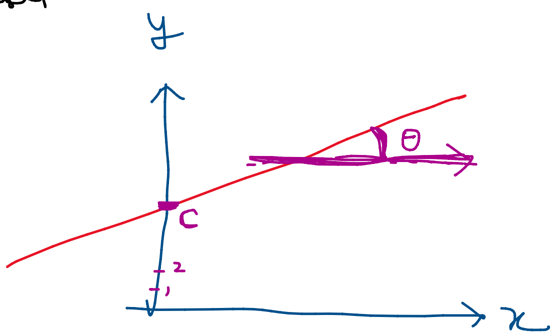

Tasks
Algorithm
Machine Learning
Supervised Learning
Regression
Linear Regression
Classification
Logistic Regression
Unsupervised learning
Clustering
K-means

learning

Linear Regression Algorithm

$$y = m \times x + c$$

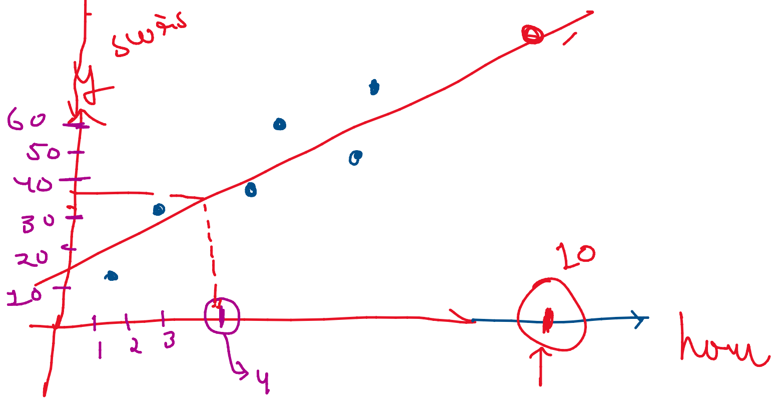
(m ← slope, c ← intercept)



$m = \tan\theta$

hours (no. of hours) → x (input)	Scores (Target Variable) → y (output)
5	50
3	25
8	90
⋮	⋮

Target Variable → Continuous → Regression → Linear Regression



$\{ scores = m * hours + c \}$

$\rightarrow score = m * 10 + c$

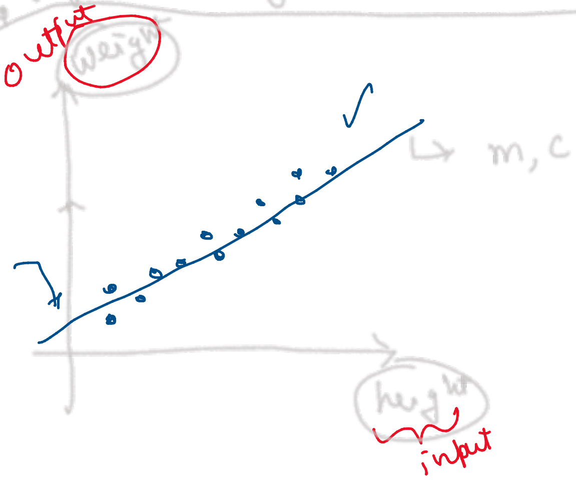
$\{ D_n \rightarrow$ Linear Reg $\rightarrow m, c \}$

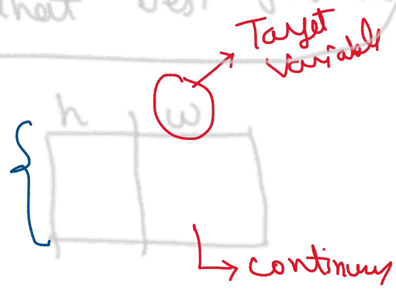
Prediction

input → $m * input + c$ → $\hat{y}$

Linear Reg — I

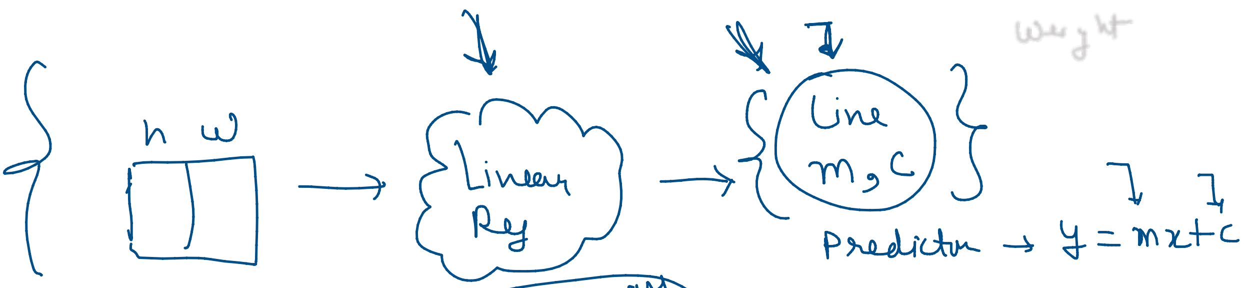
Task → Trying to find a line that best fits the data

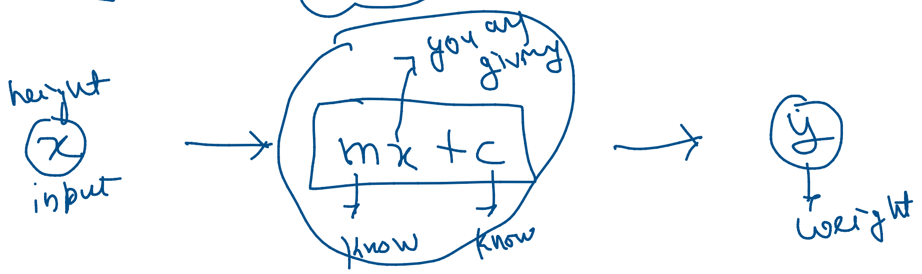




$y = m x + c$

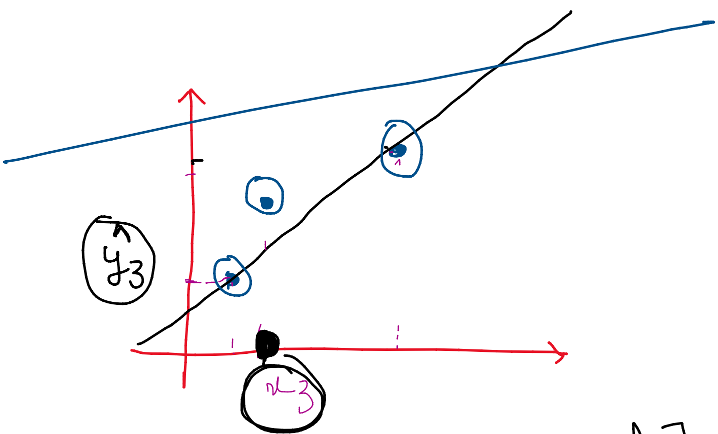
y → weight, x → height





Logist

Task → To find a line that 'best fits' the given data



1. error → 0
2. error → 0
3. error → $y_3 - \hat{y}_3$ (actual y_3, pred $\hat{y}_3$)

Total I

[illegible]

Total →

↳ White line → $0 + 0 + y_3 - \hat{y}_3$ → error less

Blue line → $e_1 + e_2 + e_3$ → more error

x_i

20 min

l_i = actual $- y_i$ − pred $- y_i$

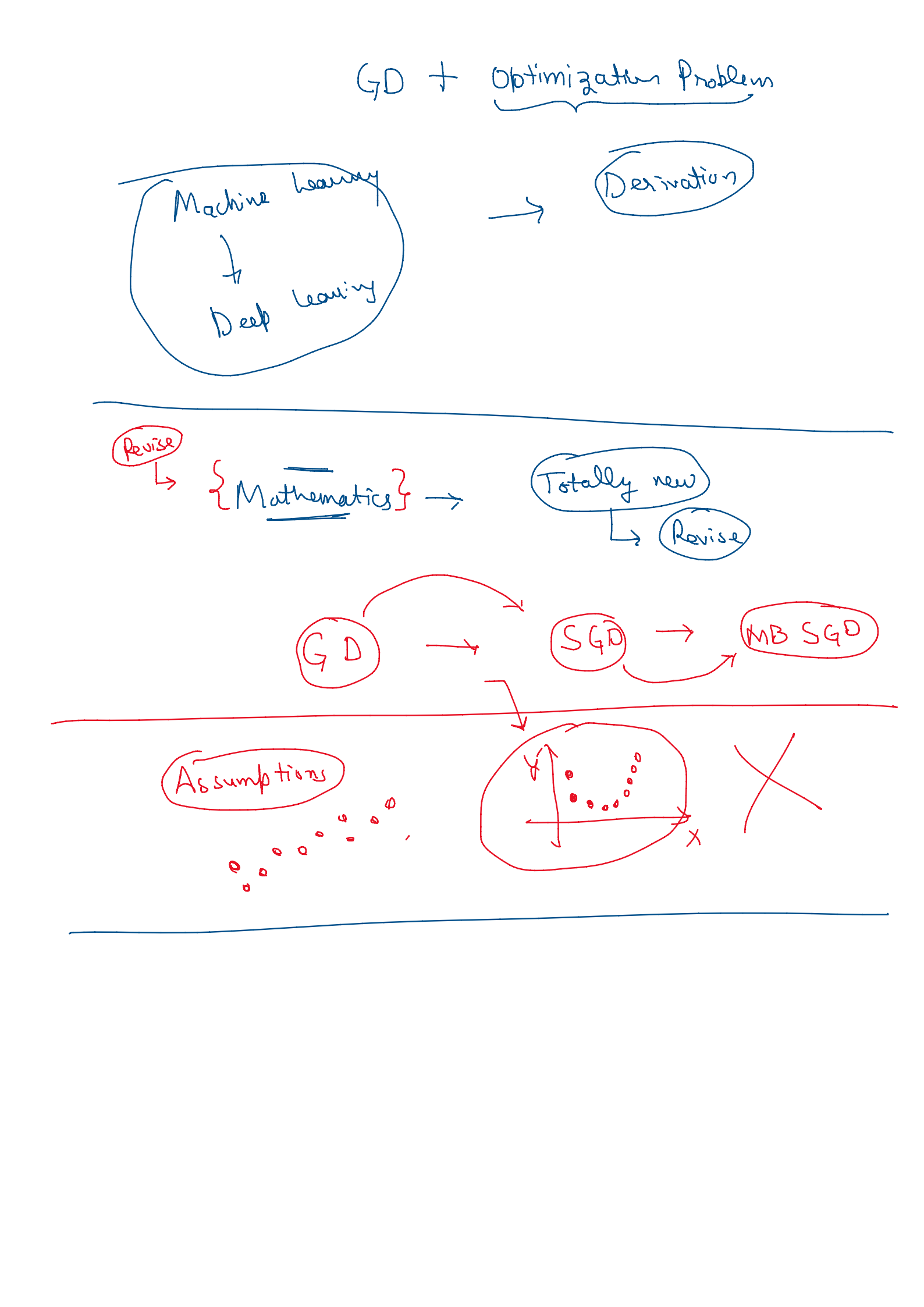
$m^*, c^* = \underset{m,c}{\arg\min} \sum_{i=1} l_i$ → Optimisation Theory

equation

differentiation → minimizing

{ Linear Regression → Ordinary least Square }

Gradient Descent → Solving Linear Reg optimization problem


GD + Optimization Problem
Machine learning
Deep learning
Derivation
Revise
{Mathematics}
Totally new
Revise
GD
SGD
MB SGD
Assumptions
Y
X