MACHINE LEARNING

Lecture : 2

DATA - PREPROCESSING

1) Data aggregation
2) Data cleaning
3) Instances selection
4) Feature taning.

① Data aggregation

KEY	VALUE
A	1
A	3
B	5
B	6
C	2
C	4

→

KEY	VALUE
A	4
B	11
C	6

Add (Aggregation)

② Data cleaning

a) Removal of data :

* Removing records of corrupted or invalid values

- Noisy : Eg: Salary = - 20
- Inconsistent : Eg: Age = 50, DOB = 1/1/2020
- Intentional : Eg: All DOB as 'Jan-2021'.

* Removing records with missing data in large number of columns

* Removing duplicate data

b) Imputing values :

* Replacing blank values with

1) Null
2) Mean / Median / Mode
3) Interpolation
4) Forward / backward fill.

c) outliers :

Method 1 : IQR

Lower limit = $Q_1 - 1.5IQR$

Upper limit = $Q_3 + 1.5IQR$

Method 2 : Z-score method

Lower limit = $\mu - 3\sigma$

Upper limit = $\mu + 3\sigma$

$\mu \rightarrow$ mean, σ - standard deviation.

③ Data partitioning

⇒ Challenge : Non-representative training data

Solution : * Increase sample size

* Correct sampling process.

Sampling methods :

* Simple random sample
* Stratified sampling
* Clustered sampling

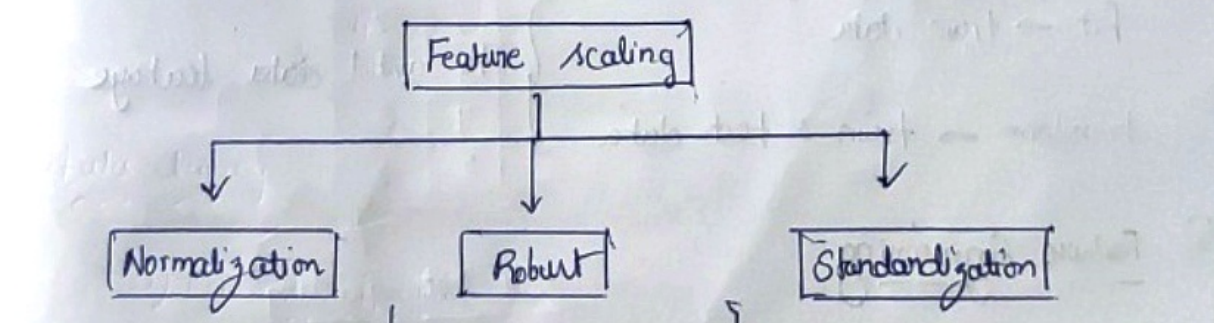
⇒ Challenge : Imbalance dataset

Solution : Under sample the majority class

(or)

Over sample the minority class

④ Feature tuning

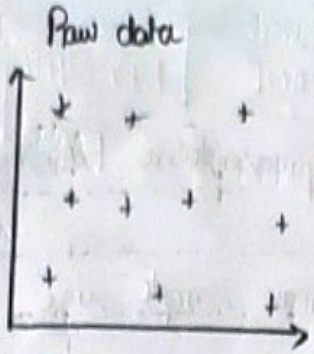


$X_N = \frac{X - X_{min}}{X_{max} - X_{min}}$

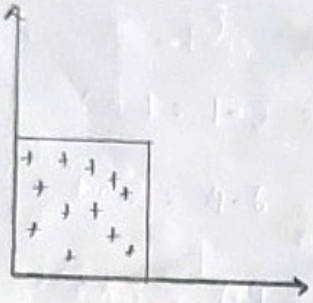
$[0, 1]$

$X_R = \frac{X - X\_median}{IQR}$

$X_S = \frac{X - mean}{Standard\ deviation}$

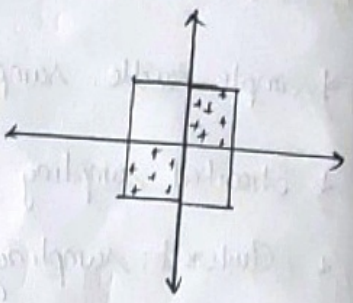


Normalization



* When the approximate upper & lower bound are known
* Approximately uniform distributed

Standardization



* Not bounded by range
* less affected by outliers

Note:

Fit → train data

transform → train & test data } to avoid data leakage

⑤ Feature Engineering

a) Feature extraction:

Reducing the number of features by creating lower dimension.

Dimension reduction techniques → PCA

b) Feature Selection :

CID	Name	Age	DoB	Height
1001	A	24	1/1/1999	175
1002	B	25	1/12/2000	-
1003	C	24	2/4/2000	-
1004	D	22	1/5/2002	-

⇒ Handling redundant feature

Age, DoB

⇒ Remove irrelevant data

CID

⇒ Dropping feature (Large data missing)

Height

c) Feature Construction :

Generating new features by using techniques

→ Polynomial Expansion

→ Feature Crossing

→ Based on domain knowledge

d) Feature transformation :

1) Numerical features :

By discretization, converting continuous data in discrete data

Binning Eg : 0-10, 10-20 / High, low, medium

In binning width can be calculated by

$$W = \frac{Max - Min}{N}$$ → N - number of bins

Eg: 1, 2, 3, 4, 5, 6, 7, 8, 9

Max = 9, Min = 1, N = 3

$$W = \frac{9-1}{3} = \frac{8}{3} = 2.66$$

Bin 1 = 1-3 = {1, 2, 3}

Bin 2 = 4-6 = {4, 5, 6}

Bin 3 = 7-9 = {7, 8, 9}

2) Categorical features:

By utilizing label encoding or one-hot encoding

Eg:

ID	Fuel
1	A
2	B
3	A
4	C
5	B

Label encoding

ID	Fuel
1	1
2	2
3	1
4	3
5	2

One-hot encoding

ID	A	B	C
1	1	0	0
2	0	1	0
3	1	0	0
4	0	0	1
5	0	1	0

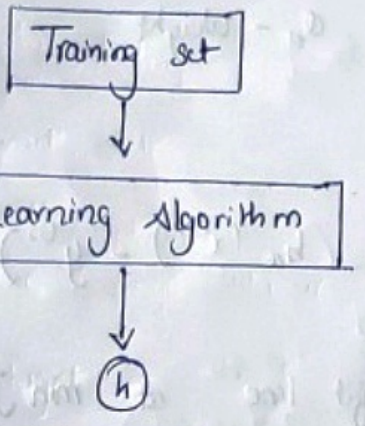
Drop the 'C' Column to avoid data redundancy ⟸

Lecture 3 : Linear regression

Linear regression :

Data $X = \{x^{(1)}, x^{(2)} \dots x^{(n)}\}$ where $x^{(i)} \in \mathbb{R}^d$

Labels $Y = \{y^{(1)}; \dots y^{(n)}\}$ where $y^{(i)} \in \mathbb{R}$

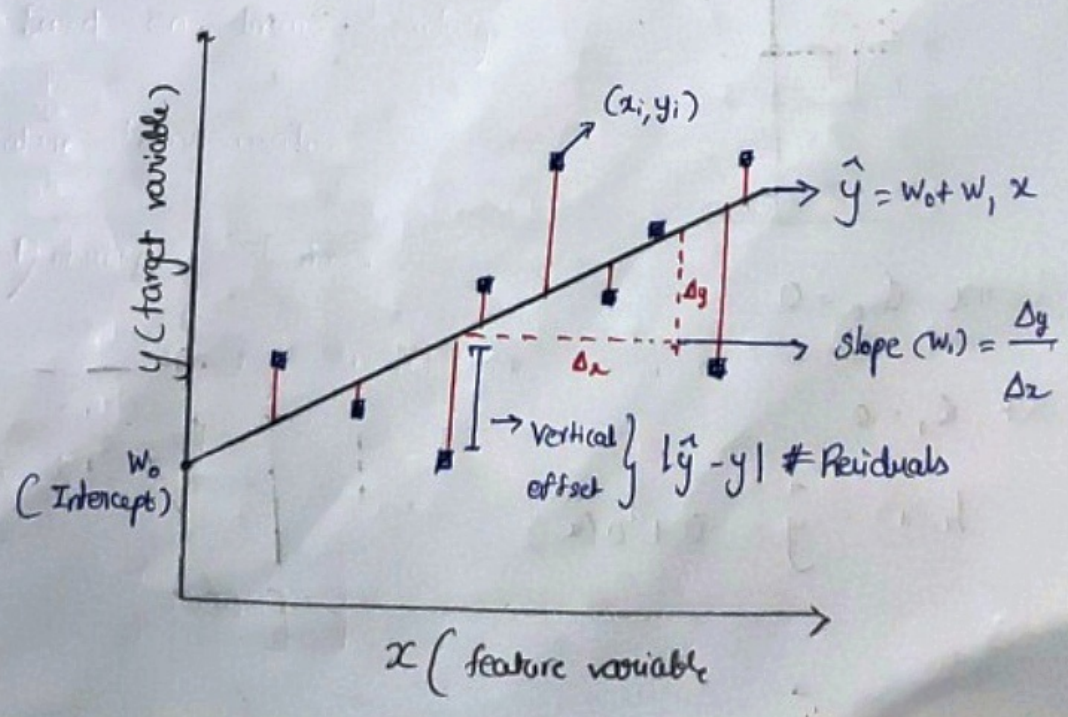


Hypothesis : $h_\theta(x) = \theta_0 + \theta_1 x$

θ_i = Parameters

h = Model

Least square regression line :



Hypothesis :

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots \theta_d x_d$$

$$\boxed{y = \sum_{i=0}^{d} \theta_i x_i} \quad \text{Assuming } x_0 = 1$$

$$\boxed{h(x) = \theta^T x}$$

Here, θ_0 - Bias, $\theta_1 \dots \theta_d$ - Weight

Cost function :

$$J(\theta) = \frac{1}{2n} \sum_{i=1}^{n} (h_\theta(x^{(i)}) - y^{(i)})^2$$

We can find the best fit line at min $J(\theta)$

$$J(\theta) = \frac{1}{2n} \sum_{i=1}^{n} (\hat{y}^{(i)} - y^{(i)})^2$$

Example :

x	y
0	0
1	1
2	2
3	3

Assume $\theta_0 = 0$

At $\theta_1 = 0$

$$h(x) = \hat{y} = 0 + 0(x)$$

At $x=0$, $x=1, 2 \cdots 3$

$\downarrow$ $y=0$ $\quad$ $\downarrow$ $y=0$

x	y	$\hat{y}$
0	0	0
1	1	0
2	2	0
3	3	0

$$J(\theta) = \frac{1}{2(4)} \left[(0-0)^2 + (1-0)^2 + (2-0)^2 + (3-0)^2 \right]$$

$$\boxed{J(\theta) = 1.75}$$

At $\theta_1 = 0.5 \Rightarrow \hat{y} = 0 + 0.5(x)$

$x=0$, $x=1$, $x=2$, $x=3$

$\downarrow$ $y=0$ $\quad$ $\downarrow$ $y=0.5$ $\quad$ $\downarrow$ $y=1$ $\quad$ $\downarrow$ $y=1.5$

x	y	$\hat{y}$
0	0	0
1	1	0.5
2	2	1
3	3	1.5

$$J(\theta) = \frac{1}{8} \left[(0-0)^2 + (1-0.5)^2 + (2-1)^2 + (3-1.5)^2 \right]$$

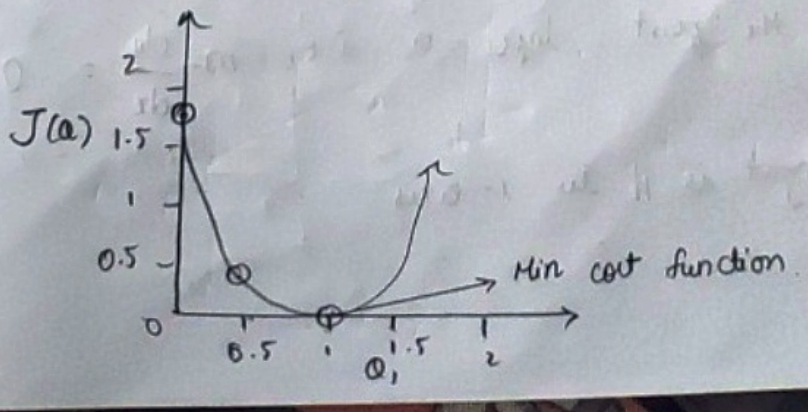
$$\boxed{J(\theta) = 0.43}$$

At $\theta_1 = 1 \Rightarrow \hat{y} = 0 + 1x$

x	y	$\hat{y}$
0	0	0
1	1	1
2	2	2
3	3	3

$$J(\theta) = \frac{1}{8}(0)$$

$$\boxed{J(\theta) = 0} \quad [\min]$$



Important points

$$h(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x_i$$

Parameters to learn → θ_i (Bias and weight)

$$\text{Cost function} = \frac{1}{2n} \sum_{i=1}^{n} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2 = J(\theta)$$

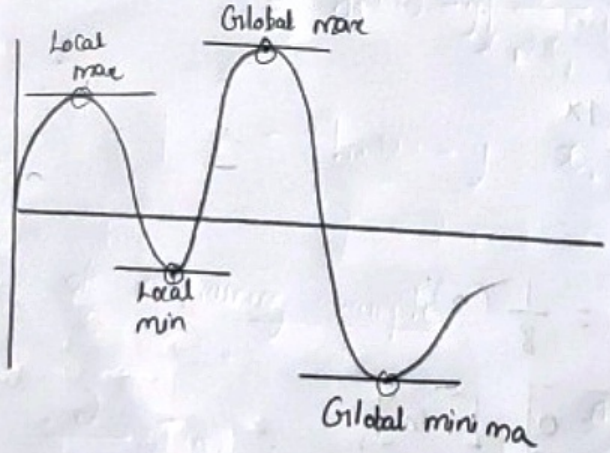
Goal → $\min_{\theta} (J(\theta))$

Gradient descent

$$f(x,y) = x^2 y + \sin(y)$$

$$\frac{\partial f}{\partial x} = 2xy \quad , \quad \frac{\partial f}{\partial y} = x^2 + \cos(y)$$

$$\nabla f(x,y) = \begin{bmatrix} 2xy \\ x^2 + \cos(y) \end{bmatrix}$$



At all the point slope = 0 i.e, $m = \frac{dy}{dx} = 0$

tangent is ∥ to x-axis

Algorithm

① Initialize θ

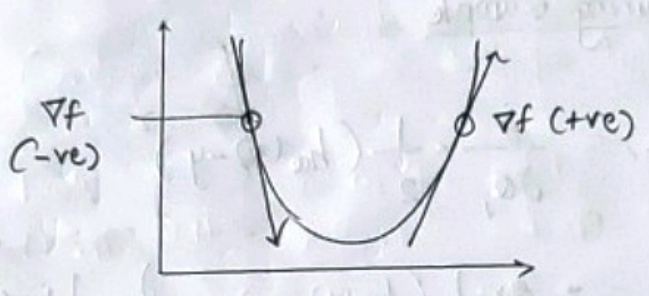
② Repeat until convergence

$$\theta_j = \theta_j^{old} - \alpha \left[\frac{\partial}{\partial \theta_j} J(\theta) \right]$$

→ gradient of cost fn w.r.t to θ

θ_j^{old} → Old weight

α → Learning rate



Case (i) : $\nabla f = -ve$

$$\theta_j = \theta_j^{old} - 0.5 (-ve \text{ value}) = \theta_j^{old} \oplus 0.5x$$

θ_j will increase

Case (ii) $\nabla f = +ve$

$$\theta_j = \theta_j^{old} - 0.5 (+ve \text{ value}) = \theta_j^{old} \ominus 0.5x$$

θ_j will decrease

Learning rate :

1) α is small

Takes longer time to converge

2) α is large

Overshoot the θ_1 value and fail to converge

Derivation of cost function

$$J(\theta) = \frac{1}{2n} \sum_{i=1}^{n} (h_\theta(x^{(i)}) - y^{(i)})^2$$

Differentiating w.r.t θ_j

$$\frac{\partial}{\partial \theta_j}(J(\theta)) = \frac{\partial}{\partial \theta_j} \times \frac{1}{2n} \sum_{i=1}^{n} (h_\theta(x^{(i)}) - y^{(i)})^2$$

Derivative of one training example ($n=1$)

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{\partial}{\partial \theta_j} \times \frac{1}{2} (h_\theta(x) - y)^2$$

$$= 2 \times \frac{1}{2} (h_\theta(x) - y) \cdot \frac{\partial}{\partial \theta_j} [h_\theta(x) - y]$$

we know, $\boxed{h_\theta(x) = \sum_{i=0}^{n} \theta_i x_i}$

$$= [h_\theta(x) - y] \frac{\partial}{\partial \theta_j} \left(\sum_{i=0}^{n} \theta_i x_i - y \right)$$

$$\boxed{\theta_1 = (h_\theta(x) - y)\, x_i}$$

Derivative of n-training example

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{\partial}{\partial \theta_j} \times \frac{1}{2n} \sum_{i=1}^{n} \left[h_\theta(x^{(i)}) - y^{(i)} \right]^2$$

$$= \frac{\partial}{\partial \theta_j} \times \frac{1}{2n} \sum_{i=1}^{n} \left[\sum_{k=0}^{d} \theta_k x_k^{(i)} - y^{(i)} \right]^2$$

$$= \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{k=0}^{d} a_k x_k^{(i)} - y^{(i)} \right) \times \frac{\partial}{\partial a_j} \left(\sum_{k=0}^{d} a_k x_k^{(i)} - y^{(i)} \right)$$

$$\boxed{a_j = \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{k=0}^{d} a_k x_k^{(i)} - y^{(i)} \right) \times x_j^{(i)}}$$

Gradient descent - variants

1) Batch gradient

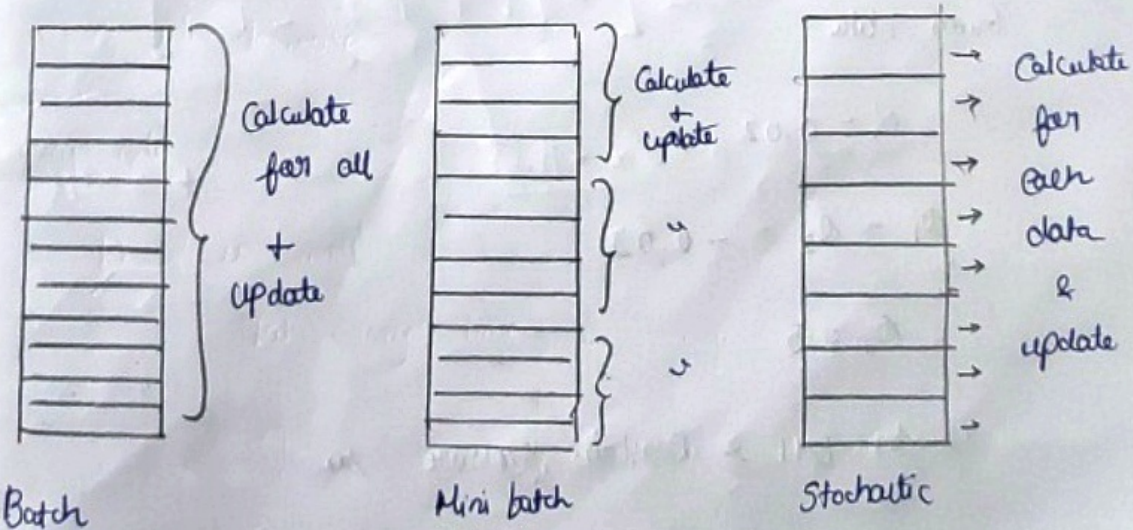
Calculating the derivative from all the training data before calculating an update.

2) Mini-Batch gradient

Calculating the derivative of mini grop of training data before calculating the update

3) Stochastic gradient

Calculating the derivative from each training instance and update immediately



Problem:

Fit a linear regression. Show only the first iteration of gradient descent algorithm using learning rate of 0.02 for the following data.

If the relative risk of coronary heart disease is believed only linearly dependant on BMI as well as diastolic pressure. Assume the intercept of the regression model as 5 and the slope of independant variables as −0.03 (negative)

Patient	Systolic Pressure (mm Hg)	Diastolic Pressure (mm Hg)	BMI	Waist Threshhold cm	RR - CHD [Relative risk of chronic heart disease]
1	140	80	35	100	1.81
2	120	80	25	80	1.22
3	130	100	30	60	1.71
		x_1	x_2		y

Solution:

Given data:

$$\alpha = 0.02$$

$$\theta_1 = \theta_2 = -0.03$$

$$\theta_0 = 5$$

Consider BMI & Diastolic pressure as

x_1 = Diastolic pressure, x_2 = BMI

Steps:

① Identification of equation

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x$$

$$\boxed{RA\text{-}CHD = 5 - (0.03 \times \text{Diastolic pressure}) - (0.03 \times BMI)}$$

② Find the new parameters

$$\theta_j = \theta_j^{old} - \frac{1}{n}\alpha \sum_{i=1}^{n} \left(h_\theta(x^{(i)}) - y^{(i)}\right) \times x_j^{(i)}$$

Intercept

$$\theta_0^{new} = \theta_0^{old} - \frac{1}{3} \times 0.02 \sum_{i=1}^{3} \left[(5 - 0.03x_1 - 0.03x_2) - y\right]$$

$$= 5 - (0.006) \begin{cases} (5 - 0.03(80) - 0.03(35)) - 1.81 \\ \text{.....} \quad (+) \\ (5 - 0.03(80) - 0.03(25)) - 1.22 \\ \quad (-) \\ (5 - 0.03(100) - 0.03(30)) - 1.71 \end{cases}$$

$$= 5 - 0.006\left[(1.55 - 1.81) + (1.85 - 1.22) + (1.1 - 1.71)\right]$$

$$= 5 - 0.006\left[-0.26 + 0.63 - 0.61\right]$$

$$= 5 - 0.006[-0.24]$$

$$= 5 + 0.00144$$

$$\boxed{\theta_0 = 5.00144}$$

Weight (θ_1)

$$\theta_1^{new} = \theta_1^{old} - \frac{1}{R} \times 0.02 \sum_{i=1}^{3} \{[5 - 0.03x_1 - 0.03x_2] - y\} x_1$$

$$= -0.03 - 0.006 \begin{cases} [(5 - 0.03(80) - 0.03(35)) - 1.81]\,80 \\ + \\ [(5 - 0.03(80) - 0.03(25)) - 1.92]\,80 \\ + \\ [(5 - 0.03(100) - 0.03(30)) - 1.71]\,100 \end{cases}$$

$$= -0.03 - 0.006 \left[(-0.26 \times 80) + (0.63 \times 80) + (-0.61 \times 100) \right]$$

$$= -0.03 - 0.006 \left[-20.8 + 50.4 - 61 \right]$$

$$= -0.03 - 0.006 (-31.4) \quad = -0.03 + 0.1884$$

$$\theta_1^{new} = 0.1584$$

Weight (θ_2)

$$\theta_2^{new} = -0.03 - \frac{0.02}{3} \left[(-0.26 \times 35) + (0.63 \times 25) + (-0.61) \times 30 \right]$$

$$= -0.03 - \frac{0.02}{3} \left[-9.1 + 15.75 - 18.3 \right]$$

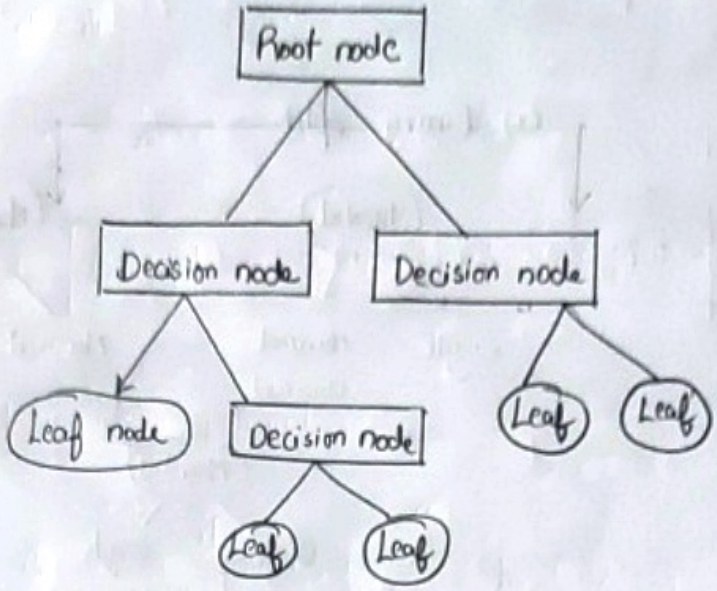
$$= -0.03 - \frac{0.02 \times -11.65}{3} \quad = -0.03 + 0.0776$$

$$\theta_2^{new} = 0.0477$$

Updated parameters after iteration ①

$\theta_0 = 5.00144$, $\theta_1 = 0.1584$, $\theta_2 = 0.0477$

Lecture : 7 & 8 Decision tree



A flow chart with tree like structure

→ Internal node denotes the test on attribute

→ Leaf node represents class labels

Two phases

1) Construction :

- All the records will be at the root
- Partition the records recursively based on the attribute conditions

2) Pruning :

Identify and remove branches that reflects noise or outliers

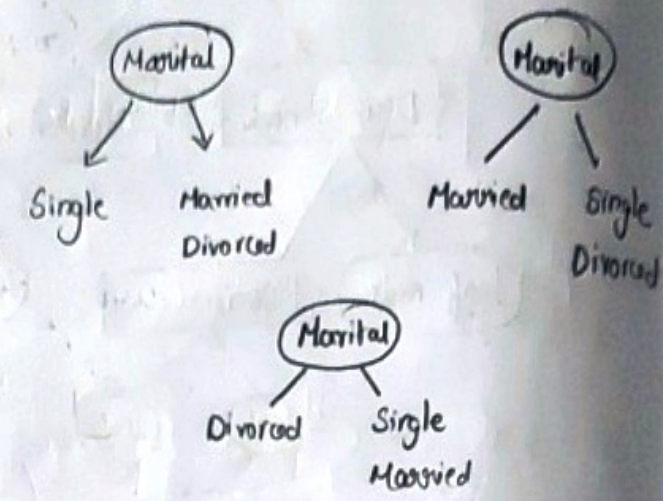
* Pre pruning
* Post pruning.

Node split for different attribute types

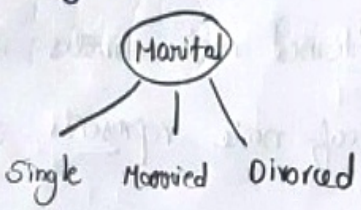
1) Nominal :

Marital Status
Single
Married
Divorced

(i) Binary split



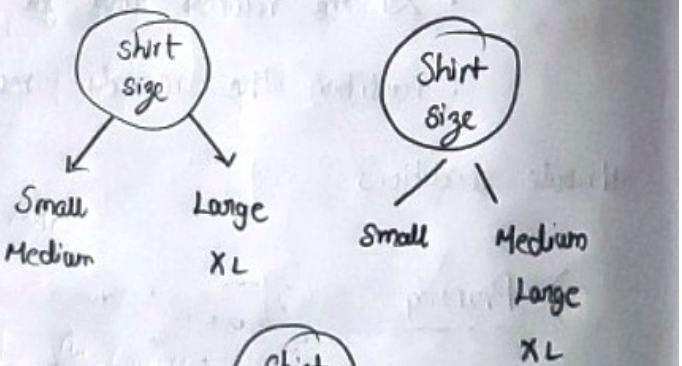
(ii) Multi way split



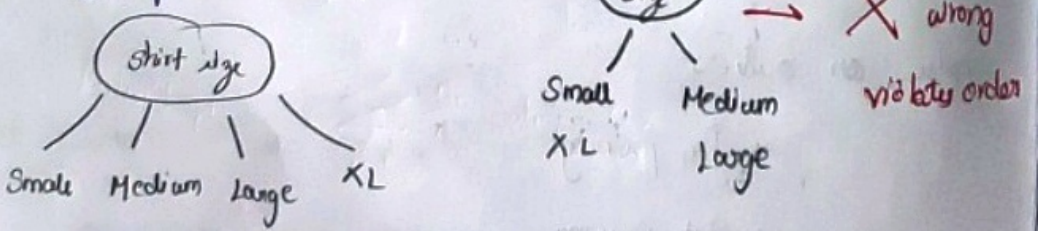
2) Ordinal :

Shirt size
Small
Medium
Large
XL

(i) Binary split



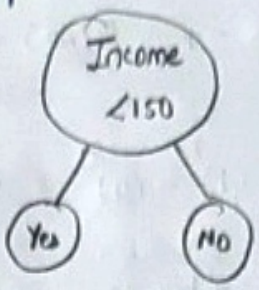
(ii) Multi way split



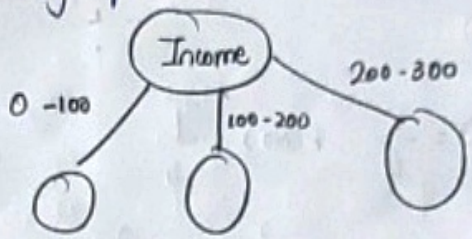
3) Continuous

Salary
100
200
300
250
120
140
80

(i) Binary split



(ii) Multiway split



Measure of node impurity

Gini Index

$$\text{Gini Index} = 1 - \sum_{i=0}^{C-1} P_i(t)^2$$

(0 - 0.5)

$P_i \rightarrow$ Probability of class i

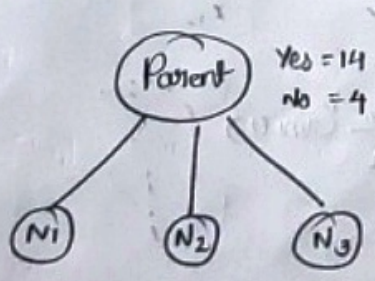
$C \rightarrow$ total number of class

Entropy

$$\text{Entropy} = -\sum_{i=0}^{C-1} P_i(t) \log_2 P_i(t)$$

(0 - 1)

Example :



N_1	N_2	N_3
Yes = 6	Yes = 5	Yes = 3
No = 0	No = 1	No = 3

At node ① :

$$P(Yes) = \frac{6}{6} = 1, \quad P(No) = \frac{0}{6} = 0$$

$$\text{Gini} = 1 - (1)^2 - (0)^2 = 0$$

$$\text{Entropy} = -0 \log_2(0) - 1 \log_2(1) = 0$$

At node ②

$$P(Yes) = \frac{5}{6}, \quad P(No) = \frac{1}{6}$$

$$\text{Gini} = 1 - \left(\frac{5}{6}\right)^2 - \left(\frac{1}{6}\right)^2 = 1 - \frac{25}{36} - \frac{1}{36} = 0.278$$

$$\text{Entropy} = -\frac{5}{6}\log_2\left(\frac{5}{6}\right) - \frac{1}{6}\log_2\left(\frac{1}{6}\right) = 0.650$$

At node ③

$$P(Yes) = \frac{3}{6}, \quad P(No) = \frac{3}{6}$$

Gini = 0.5

Entropy = 1

Note if both P(i) is same it is highly impure ∴ Gini = 0.5, Entropy = 1

Impurity for collection of node

$$\text{Gini}_{split} = \sum_{i=1}^{K} \frac{n_i}{n} \text{Gini}(i)$$

$$\text{Entropy}_{split} = \sum_{i=1}^{K} \frac{n_i}{n} \text{Entropy}(i)$$

n = number of records at parent node

n_i = number of records at child

From the previous calculation

$$Gini_{split} = \frac{6}{18} \times Gini\,① + \frac{6}{18} Gini\,② + \frac{6}{18} \times Gini\,③$$

$$= \left(\frac{6}{18} \times 0\right) + \left(\frac{6}{18} \times 0.278\right) + \left(\frac{6}{18} \times 0.5\right)$$

$$\boxed{Gini_{split} = 0.259}$$

$$Entropy_{split} = \frac{6}{18} Entropy\,① + \frac{6}{18} Entropy\,② + \frac{6}{18} Entropy\,③$$

$$= \left(\frac{6}{18} \times 0\right) + \left(\frac{6}{18} \times 0.65\right) + \left(\frac{6}{18} \times 1\right)$$

$$\boxed{Entropy_{split} = 0.416}$$

Computing information gain

$$\boxed{Gain_{split} = Entropy(P) - \sum_{i=1}^{K} \frac{n_i}{n} Entropy(i)}$$

where,

P – Parent node

n_i – number of records in child node

n – Total number of records in parent

K – Number of split / partition.

Problem

Day	Outlook	Temperature	Humidity	Wind	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Step 1 : Root node selection

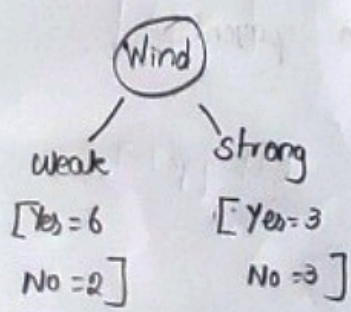
$$\text{Entropy}(P) = -P_+ \log(P_+) - P_- \log(P_-)$$

Yes = 9, No = 5

$$\text{Entropy}(P) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right)$$

$$\boxed{\text{Entropy}(P) = 0.94}$$

(i) Wind = {Weak, strong}



$$\text{Entropy}_{\text{strong}} = 1$$

$$\text{Entropy}_{\text{weak}} = -\frac{6}{8}\log_2\left(\frac{6}{8}\right) - \frac{2}{8}\log_2\left(\frac{2}{8}\right)$$

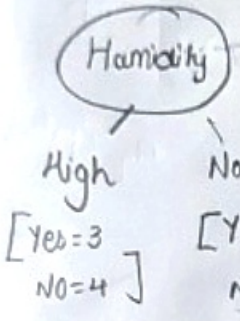
$$= 0.811$$

$$\text{Gain [Wind]} = \text{Entropy (P)} - \frac{8}{14}\text{Entropy}_{\text{weak}} - \frac{6}{14}\text{Entropy}_{\text{strong}}$$

$$= 0.94 - \frac{8}{14}(0.811) - \frac{6}{14}(1)$$

$$\boxed{\text{Gain [Wind]} = 0.048}$$

(ii) Humidity = {High, Normal}



$$\text{Entropy}_{\text{High}} = -\frac{3}{7}\log_2\left(\frac{3}{7}\right) - \frac{4}{7}\log_2\left(\frac{4}{7}\right)$$

$$= 0.985$$

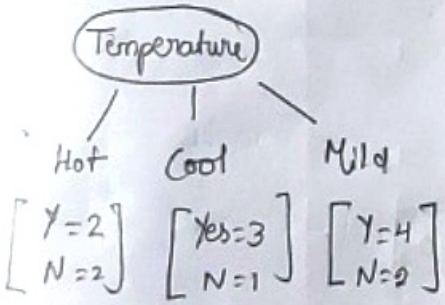
$$\text{Entropy}_{\text{Normal}} = -\frac{6}{7}\log_2\left(\frac{6}{7}\right) - \frac{1}{7}\log_2\left(\frac{1}{7}\right)$$

$$= 0.592$$

$$\text{Gain [Humidity]} = 0.94 - \frac{7}{14}(0.985) - \frac{7}{14}(0.592)$$

$$\boxed{\text{Gain [Humidity]} = 0.151}$$

(iii) Temperature = {Hot, mild, cool}



$$\text{Entropy}_{\text{Hot}} = 1$$

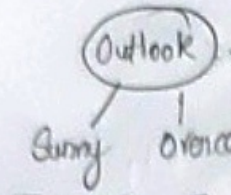
$$\text{Entropy}_{\text{Cool}} = -\frac{3}{4}\log\left(\frac{3}{4}\right) - \frac{1}{4}\log\left(\frac{1}{4}\right) = 0.622$$

$$\text{Entropy}_{\text{Mild}} = -\frac{4}{6}\log\left(\frac{4}{6}\right) - \frac{2}{6}\log\left(\frac{2}{6}\right) = 0.918$$

$$\text{Gain [Temperature]} = 0.94 - \frac{4}{14}(1) - \frac{4}{14}(0.622) - \frac{6}{14}(0.918)$$

$$\boxed{\text{Gain [Temperature]} = 0.08}$$

CIV) Outlook = {Sunny, Overcast, rain}



$$\text{Entropy}_{\text{Sunny}} = -\frac{3}{5}\log_2\left(\frac{3}{5}\right) - \frac{2}{5}\log_2\left(\frac{2}{5}\right)$$

$$= 0.97$$

$$\text{Entropy}_{\text{Overcast}} = 0$$

$$\text{Entropy}_{\text{rain}} = 0.97$$

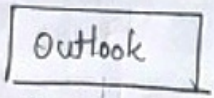
$$\text{Gain [outlook]} = 0.94 - \frac{5}{14}(0.97) - 0 - \frac{5}{14}(0.97)$$

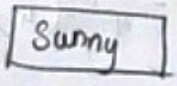
$$\boxed{\text{Gain [outlook]} = 0.247}$$

Gain	value
Wind	0.048
Hum	0.151
Temp	0.08
Outlook	0.247 ✓

Selecting outlook since it has more gain

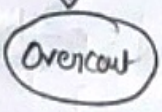
Step 2: Initial tree - ①





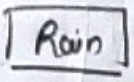
[D1, D2, D8, D9, D11]

[Yes - 2, No - 3]



Leaf

(Yes)



[D4, D5, D6, D10, D14]

[Y - 3, N - 2]

Entropy (P) = 0.97
Sunny

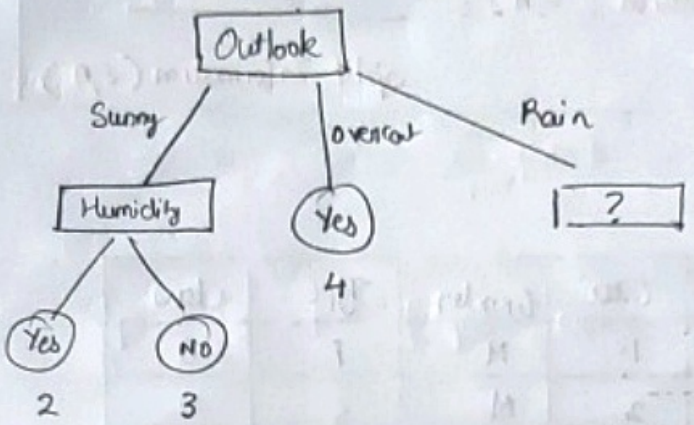
Entropy (P) = 0.97
rain

Step 3 : Finding the best node split for Sunny

Gain [Sunny, Humidity] = $0.97 - \frac{3}{5}(0) - \frac{2}{5}(0) = 0.97$

Gain [Sunny, Wind] = $0.97 - \frac{2}{5} \times 1 - \frac{3}{5} \times 0.918 = 0.019$

Gain [Sunny, Temp] = $0.97 - \frac{2}{5} \times 0 - \frac{2}{5} \times 1 - \frac{1}{5} \times 0 = 0.57$



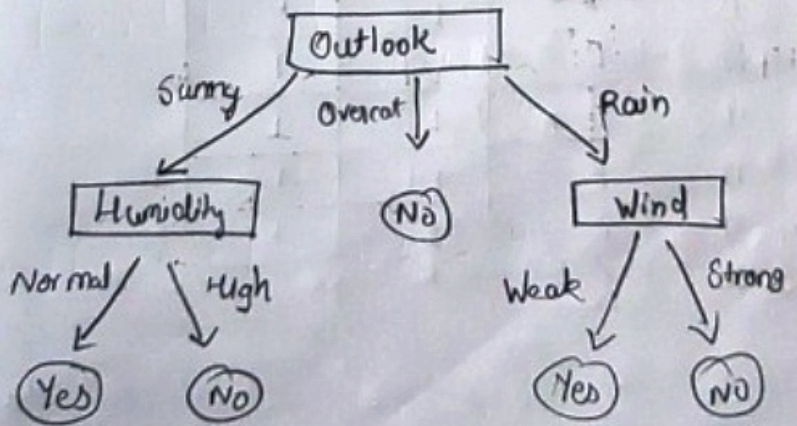
Step 4 : Find the best node split for rain

Gain [Rain, Humidity] = $0.97 - \frac{3}{5}(0.91) - \frac{2}{5}(1) = 0.024$

Gain [Rain, Wind] = $0.97 - \frac{3}{5}(0) - \frac{2}{5}(0) = 0.97$

Gain [Rain, Temp] = $0.97 - \frac{3}{5}(0.91) - \frac{2}{5}(1) = 0.024$

Step 5 : Final tree



Gain ratio

Gain ratio is an alternative measure to tackle the splits of n classes.

Eg: Customer ID will have 'n' splits with less impurity

$$\text{Gain ration}(S,A) = \frac{\text{Gain}(S,A)}{\text{Split Information}(S,A)}$$

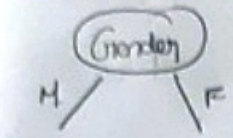
Problem :

CID	Gender	Type	class
1.	M	F	0
2	M	S	0
3	M	S	0
4	M	S	0
5	M	S	0
6	M	S	0
7	F	S	0
8	F	S	0
9	F	S	0
10	F	L	0
11	M	F	1
12	M	F	1
13	M	F	1
14	M	L	1
15	F	L	1
16	F	L	1
17	F	L	1
18	F	L	1
19	F	L	1
20	F	L	1

Entropy (Parent) $= -\frac{10}{20}\log_2\left(\frac{10}{20}\right) - \frac{10}{20}\log_2\left(\frac{10}{20}\right)$

Entropy (Parent) = 1

At gender as node:



6-0 6-1
4-1 4-0

Entropy$_{(M)} = -\frac{6}{10}\log_2\left(\frac{6}{10}\right) - \frac{4}{10}\log_2\left(\frac{4}{10}\right) = 0.971$

Entropy$_{(F)} = -\frac{4}{10}\log_2\left(\frac{4}{10}\right) - \frac{6}{10}\log_2\left(\frac{6}{10}\right) = 0.971$

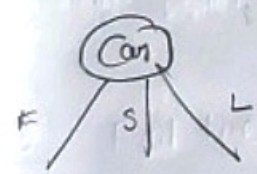
Gain [Gender] $= 1 - \frac{10}{20}(0.971) - \frac{10}{20}(0.971) = 1 - 0.971$

$= 0.029$ //

Gain ratio $= \frac{0.029}{-\frac{10}{20}\log_2\left(\frac{10}{20}\right) - \frac{10}{20}\log_2\left(\frac{10}{20}\right)} = \frac{0.029}{1}$

Gain ratio [Gender] = 0.029

At cartype as node:



1 = 0 8 = 0 1 = 0
3 = 1 7 = 1

Entropy$_{(F)} = -\frac{1}{4}\log_2\left(\frac{1}{4}\right) - \frac{3}{4}\log_2\left(\frac{3}{4}\right) = 0.811$

Entropy$_{(S)} = 0$

Entropy$_{(L)} = -\frac{1}{8}\log_2\left(\frac{1}{8}\right) - \frac{7}{8}\log_2\left(\frac{7}{8}\right) = 0.543$

Gain [type] $= 1 - \frac{4}{20}(0.811) - 0 - \frac{8}{20}(0.543)$

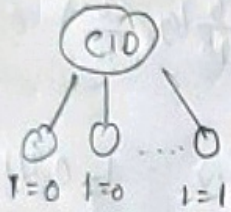
Gain [type] $= 0.620$ //

$$\text{Gain ratio [type]} = \frac{0.620}{-\frac{4}{20}\log\left(\frac{4}{20}\right) - \frac{8}{20}\log\left(\frac{8}{20}\right) - \frac{8}{20}\log\left(\frac{8}{20}\right)}$$

$$= 0.620 / 1.52$$

Gain ratio [type] = 0.41

At CID as node



$\text{Entropy}_{(CID \ldots} \quad \text{Entropy}_{(20)} = 0$

Gain [Entropy] = 1 − 0 = 1

$$\text{Gain ratio [Entropy]} = \frac{1}{\left[-\frac{1}{20}\log\left(\frac{1}{20}\right)\right] \times 20}$$

Gain ratio [Entropy] = 0.23

Summarize-

Node	Gain	Gain ratio
Gender	0.029	0.029
Type	0.62	0.41
CID	1	0.23

Even though the gain is more in the 'CID' the gain ratio is better in 'Type'. So it will be chosen to generalize the model.

Occam's razor theory :

It states that when presented with competing hypothesis that makes same prediction one should select shorter hypothesis.

Why shorter hypothesis ?

Favor : * Elegance & aesthetics
* Fewer short hypothesis than long one
* Short hypothesis are unlikely to be coincidenc

Against : * Not every short hypothesis is a reasonable one for a problem.

Handling overfitting

Overfitting in decision tree can be handled by two methods

(i) Pre-pruning
(ii) Post-pruning

Pre-pruning (Early stopping)

- Stop growing the tree earlier before it reaches the point where it perfectly classifies a data (training)

- Difficult to estimate when to stop growing the tree.

Post-pruning

Allow the tree to overfit the data and the post-prune the tree

Pre-pruning

Typical stopping condition

1) Stop if all the attributes values are same

2) Stop if all the instance belong to same class

More restrictive condition

1) Stop if the number of instance is less than user specified threshold

2) Stop expanding if the current node does not reduce impurities (Gini, Entropy)

3) Stop if generalization error < certain threshold

Problem :

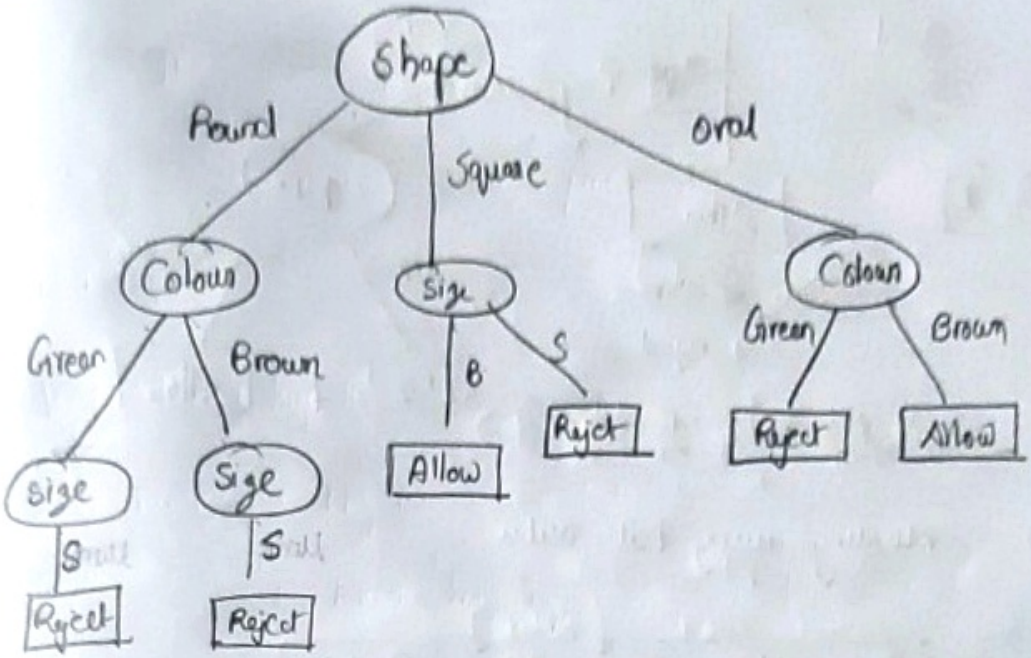
Train ⇒

Shape	Colour	Size	Action
Round	Green	S	Reject
Square	Black	B	Allow
Square	Brown	B	Allow
Round	Brown	S	Reject
Square	Green	B	Allow
Square	Brown	S	Reject
Oval	Green	B	Reject
Oval	Brown	S	Allow
Oval	Green	S	Reject

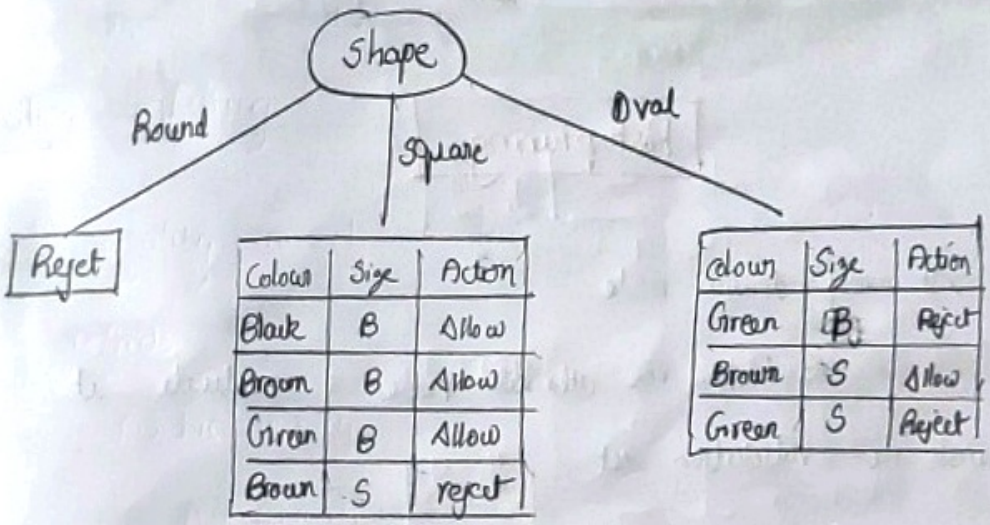
Shape	Color	Size	Action
Oval	Black	S	Reject
Round	Brown	B	Allow
Square	Brown	B	Allow
Oval	Green	S	Allow

Test

Over-fitted model



Step 1 : In shape = Round all belong to same class so we don't need to split further.



Colour	Size	Action
Black	B	Allow
Brown	B	Allow
Green	B	Allow
Brown	S	reject

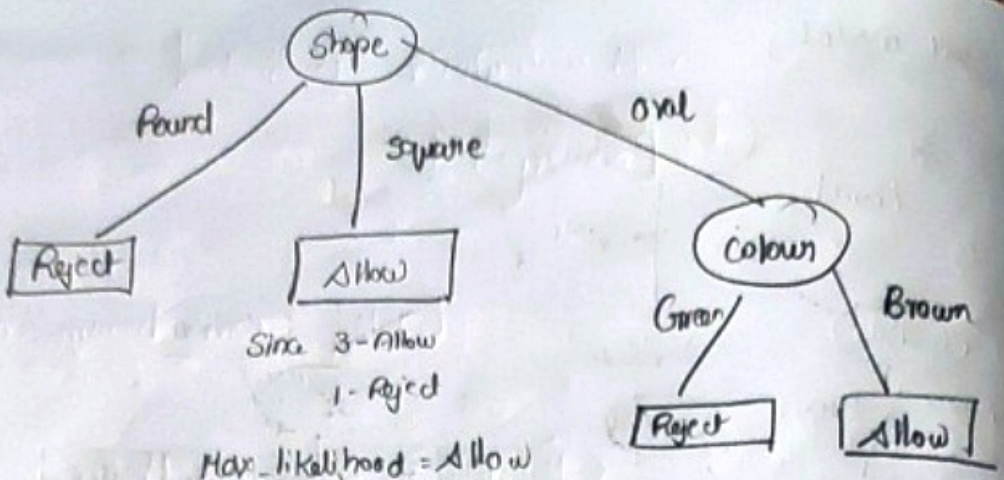
colour	Size	Action
Green	B	Reject
Brown	S	Allow
Green	S	Reject

Step 2 : Prune if maximum information gain < threshold

Assume threshold = 0.85

Max. Gain (square) = 0.8113 # <0.85 so pruning the tree.

Max. Gain (Oval) = 0.9182



Max_likelihood = Allow

Step 3 : Evaluating using test data.

Shape	Colour	Size	Action	Prediction
Oval	Black	S	Reject	—
Round	Brown	B	Allow	Reject
Square	Brown	B	Allow	Allow
Oval	Green	S	Allow	Reject

Post-pruning

At every decision node

* Retain the attribute node & evaluate it against the validation set

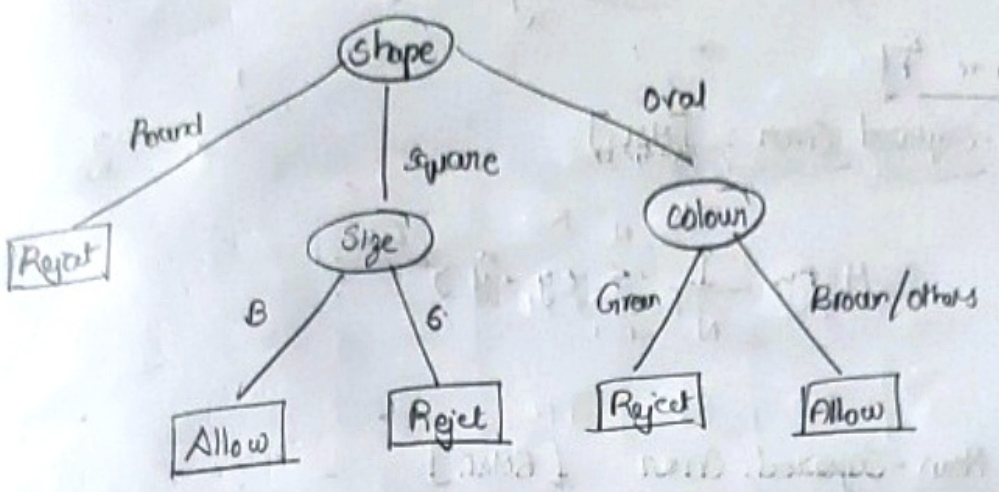
* Remove the attribute node & reevaluate it with the same prune set

* If there is a reduction in error, prune the node else retain the node

Repeat for all the branches of the tree.

Problem:

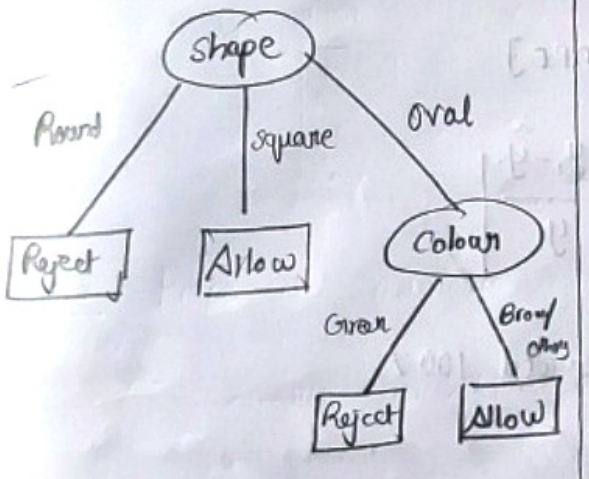
Same dataset



Shape	Color	Size	Action	Prediction
Oval	Black	S	Reject	Allow
Round	Brown	B	Allow	Reject
Square	Brown	B	Allow	Allow
Oval	Green	S	Allow	Reject

Correct prediction } = 1

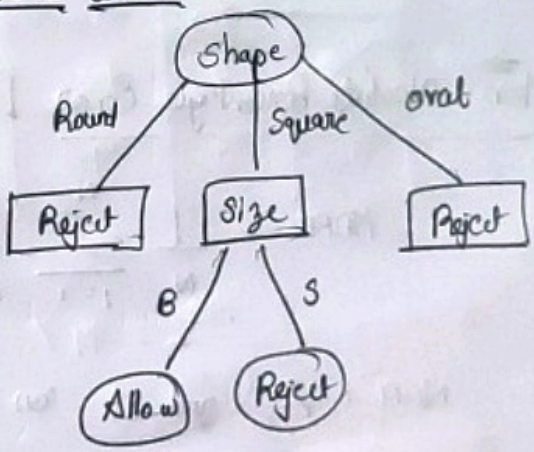
Prune Size



Action	Prediction
Reject	Allow
Allow	Reject
Allow	Allow
Allow	Reject

Correct Prediction = 1

Prune Colour ✓



Action	Prediction
Reject	Reject
Allow	Reject
Allow	Allow
Allow	Reject

Correct prediction = 2

Lecture 4: Evaluation Metrics

[Regression]

Mean-Squared Error : [MSE]

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

Root Mean-Squared Error [RMSE]

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{N} (y_i - \hat{y})^2}$$

Mean-Absolute Error [MAE]

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

Mean-Absolute Percentage Error [MAPE]

$$MAPE = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{y_i - \hat{y}}{y_i} \right|$$

MAPE = % and it can exceed 100 %.

R-Squared [R^2]

$$R^2 = 1 - \frac{SS_{residual}}{SS_{Total}}$$

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

$$SS_{Total} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

Adjusted R-Squared [R^2_{adj}]

$$R^2_{adj} = 1 - \frac{(1-R^2)(n-1)}{n-p-1}$$

where R^2 → R-squared value

P → number of predictor / feature

n → number of records in dataset

Summarized

MAE - Robust to outlier
MSE
RMSE
} Measure predictive power of model

R^2_{adj} - Compare model of different number of feature
R^2
} goodness of fit

Calculation:

y	$\hat{y}$	$\lvert y-\hat{y}\rvert$	$(y-\hat{y})^2$	$(y-\bar{y})^2$
3	2.5	0.5	0.25	2.25
4	4.5	-0.5	0.25	0.25
5	4	1	1	0.25
6	6	0	0	2.25

$\bar{y} = 4.5$

$n = 4$, $P = 1$

$MSE = 0.375$

$RMSE = \sqrt{MSE} = 0.612$

$MAE = 0.5$

$MAPE = 12.29\%$

$R^2 = 1 - 1.5/3 = 0.7$

$R^2_{adj} = 1 - \frac{(1-0.7)(4-1)}{4-1-1} = 0.55$

Lecture 6 : Evaluation metric [Classification]

Confusion Matrix

		Predicted: Yes	Predicted: No
Actual	Yes	TP	FN
	No	FP	TN

Accuracy

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + TN}, \quad \text{Error rate} = \frac{FP + FN}{TP + TN + FP + FN}$$

Problem with accuracy :

	Yes	No
Yes	0	10
No	0	990

$$\text{Accuracy} = \frac{990}{1000} = 99\%.$$

Even though accuracy is 99%. this model ignores the 'Yes' case completely.

Recall / Sensitivity / TPR :

$$\text{Recall} = \text{Sensitivity} = TPR = \frac{TP}{TP+FN}$$ | $FN \downarrow$

Specificity / TNR :

$$\text{Specificity} = TNR = \frac{TN}{TN+FP}$$

Precision :

$$\text{Precision (Yes)} = \frac{TP}{TP+FP}$$ | $FP \downarrow$

$$\text{Precision (No)} = \frac{TN}{TN+FN}$$

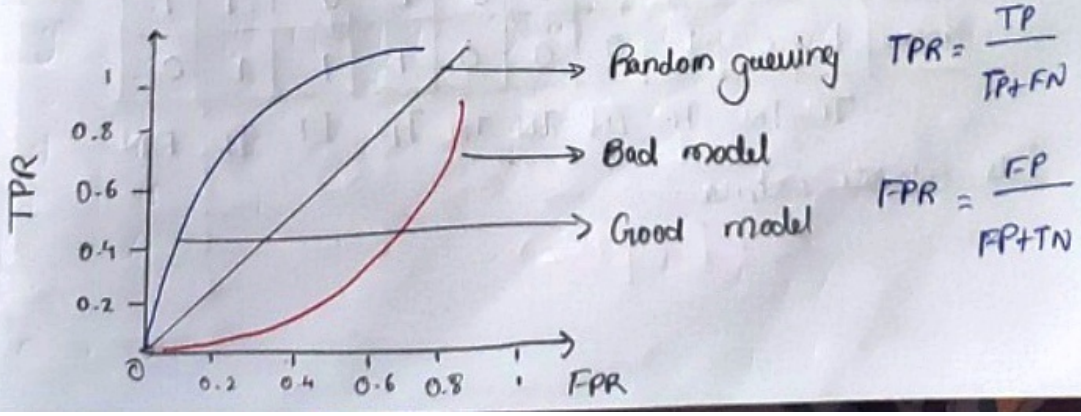
F1-Score

$$F1 = 2 \times \frac{\text{Precision} \times \text{recall}}{\text{Precision} + \text{recall}}$$

1) Suitable for class imbalance data

2) High F1 score indicates perfect precision & recall.

ROC-curve



AUC - Area Under Curve

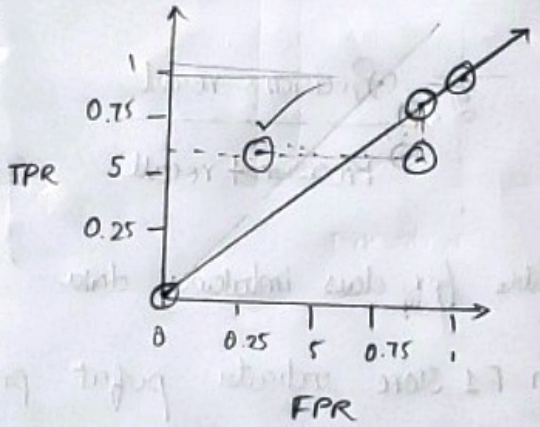
- Integrate area under the curve
- Perfect score is 1
- 0.5 - Random guessing <0.5 - Something wrong in model

Calculation of ROC

①

No	Score	Class
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

Threshold	0.25	0.5	0.75	1
TP	5	4	3	0
FP	5	4	4	0
TN	0	1	1	5
FN	0	1	2	5
TPR	1	0.8	0.6	0
FPR	1	0.8	0.8	0



②

Actual	1	0	1	1	0	0	1	0	1	1
Predicted	1	0	1	0	0	0	1	1	1	0
	TP	TN	TP	FN	TN	TN	TP	FP	TP	FN

Lets Calculate metrics

Confusion matrix

Actual \ Predicted	0	1
0	3	1
1	2	4

TP = 4, TN = 3

FP = 1, FN = 2

Accuracy:

$$\text{Accuracy} = \frac{4+3}{4+3+1+2} = \frac{7}{10}$$

Precision:

$$\text{Precision} = \frac{4}{5}$$

Recall:

$$\text{Recall} = \frac{4}{6}$$

F1 score:

$$F1 = 2 \times \frac{\left(\frac{4}{6}\right) \times \left(\frac{4}{5}\right)}{\frac{4}{6} + \frac{4}{5}} = 2 \times \frac{0.5334}{1.4667} = 0.7272$$

Specificity:

$$TNR = \frac{3}{4}$$

Lecture 5 : Logistic Regression

Generative models

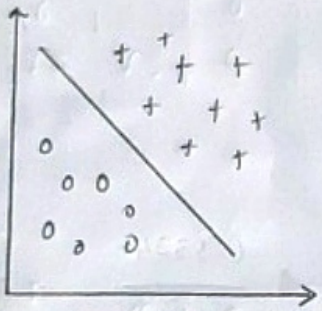
$$P(c|x) = \frac{P(x|c) \cdot P(c)}{P(x)}$$

- $P(x|c)$: Likelihood
- $P(c)$: class prior probability
- $P(x)$: Predictor prior probability
- $P(c|x)$: Posterior probability

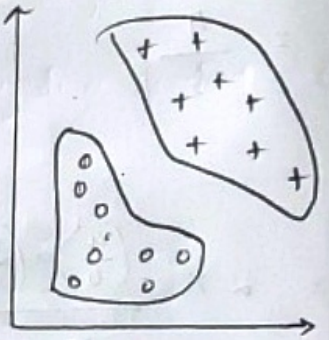
Discriminative models

$$\theta_0 + \sum_i \theta_i x_i \geq 0 \quad \text{class - 1}$$

$$\theta_0 + \sum_i \theta_i x_i < 0 \quad \text{class - 0}$$



Discriminative

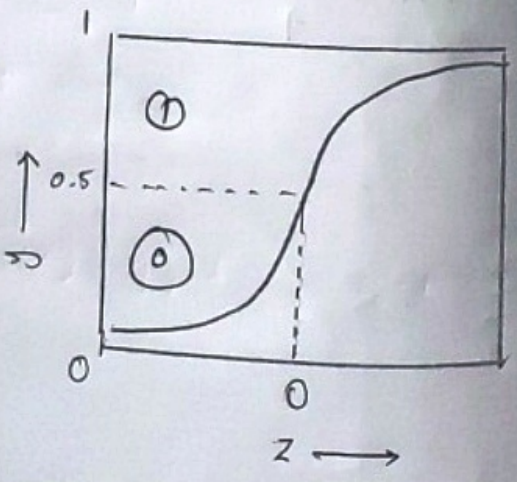


Generative

Logistic Regression

$$y(x) = \sigma(w^T x + w_0)$$

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$



let represent

$$h_w(x) = g(w^T x)$$

$$g(z) = \sigma(z) = \frac{1}{1+e^{-z}}$$

If $h_w(x) \geq 0.5$ then $y = 1$

$h_w(x) < 0.5$ then $y = 0$

Example:

$$h_w(x) = g(w_0 + w_1 x_1 + w_2 x_2)$$

If $w_0 = -3,\ w_1 = 1,\ w_2 = 1$

$$h_w(x) = g(-3 + x_1 + x_2)$$

If $-3 + x_1 + x_2 \geq 0$ then $y = 1$

i.e, $\boxed{x + x_2 \geq 3}$

$\boxed{x + x_2 < 3} \Rightarrow y = 0$

Cost function:

$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

$$\text{cost}(h_\theta(x), y) = -y \log(h_\theta(x)) - (1-y) \log(1 - h_\theta(x))$$

Gradient descent for logistic regression

$$\theta_j^{new} = \theta_j^{old} - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

Same as linear

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

Problem :

CGPA	IQ	Job
6.5	6.7	1
5	7	0
8	6	1
9	7	1
6	8	0
7.5	7.3	0

Parameters :

Learning rate = 0.3

Initial weight = (0.5, 0.5, 0.5)

Regularization constant = 0

Solution :

$$w_i^T = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

By substituting :

$$w_i^T = 0.5 + 0.5(GPA) + 0.5\,IQ$$

Calculating next θ_0

$$\theta_0^{new} = \theta_0^{old} - \frac{1}{m}(\alpha) \sum_{i=1}^{6} (h_\theta(x^i) - y^i) x_0 \qquad x_0 = 1$$

$$= 0.5 - \frac{0.3}{6}\left[\begin{array}{l}(\sigma(0.5+0.5(5.5)+0.5(6.7))-1)\\ +\\ (\sigma(0.5+0.5(5)+0.5(7))-0)\\ +\\ (\sigma(0.5+0.5(8)+0.5(6))-1)\\ +\\ (\sigma(0.5+0.5(9)+0.5(7))-1)\\ +\\ (\sigma(0.5+0.5(6)+0.5(8))-0)\\ +\\ (\sigma(0.5+0.5(7.5)+0.5(7.3))-0)\end{array}\right]$$

$$= 0.5 - \frac{0.3}{6}\left[\begin{array}{l}(\sigma(6.6)-1)+(\sigma(6.5)-0)+(\sigma(7.5)-1)+\\ (\sigma(8.5)-1)+(\sigma(7.5)-0)+(\sigma(7.9)-0)\end{array}\right]$$

$$= 0.5 - \frac{0.3}{6}\left[\begin{array}{l}(0.9986-1)+(0.9984-0)+(0.9994-1)+\\ (0.9997-1)+(0.9994-0)+(0.9996-0)\end{array}\right]$$

$$= 0.5 - \frac{0.3}{6}\left[2.9951\right]$$

$$= 0.5 - 0.149755$$

$$\boxed{\theta_0^{new} = 0.35}$$

Calculating new θ_1

$$\theta_1^{new} = \theta_1^{old} - \frac{1}{m}\alpha \sum_{i=1}^{6}\left(h_\theta(x^{(i)}) - y^{(i)}\right) x_{1\,COPA}$$

$$= 0.5 - \frac{0.3}{6}\left[\begin{array}{l}(0.9986-1)5.5+(0.9984-0)5+\\ (0.9994-1)8+(0.9997-1)9+\\ (0.9994-0)6+(0.9996-0)7.5\end{array}\right]$$

$= 0.5 - \frac{0.3}{6}(18.4702)$ $= 0.5 - 0.9235)$

$\boxed{\theta_1 = -0.42}$

Calculating new θ_2

$$\theta_2^{new} = \theta_2^{old} - \frac{1}{m}\alpha \sum_{i=1}^{6} (h_\theta(x^i) - y^{(i)})\, x_{IQ}^i$$

$$= 0.5 - \frac{0.3}{6}\begin{bmatrix} (0.9986-1)6.7 + (0.9984-0)7 + \\ (0.9994-1)6 + (0.9997-1)7 + \\ (0.9994-0)8 + (0.9996-0)7.3 \end{bmatrix}$$

$$= 0.5 - \frac{0.3}{6}(22.266)$$

$$= 0.5 - 1.1133$$

$\boxed{\theta_2 = -0.61}$

Updated equation

$$\boxed{w^T x = 0.35 - 0.42\,CGPA - 0.61\,IQ}$$

Regularized logistic regression

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)} \log(h_\theta(x^i)) + (1-y^{(i)}) \log(1-h_\theta(x^i))\right]$$

$$+ \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

↓

regularization

By taking gradient

$$\theta_i^{new} = \theta_i - \alpha \left[\left(\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_i^{(i)} \right) + \frac{\lambda}{m} \theta_i \right]$$

Lecture - 4

Closed form solution

$$\theta = (X^T X)^{-1} X^T y$$

Problem:

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

Here $X = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \end{bmatrix}$ (X_0 - Intercept) $X^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 \end{bmatrix}$

$X_0 \quad X_1$

$$y = \begin{bmatrix} 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}$$

$$\theta = \left\{ \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \end{bmatrix} \right\}^{-1} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}$$

$$\theta = \begin{bmatrix} 5 & 15 \\ 15 & 55 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}$$

$$A^{-1} = \frac{1}{\det A}[\text{Adj } A] = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

$$= \frac{1}{275-225}\begin{bmatrix} 55 & -15 \\ -15 & 5 \end{bmatrix}.$$

$$A^{-1} = \frac{1}{50}\begin{bmatrix} 55 & -15 \\ -15 & 5 \end{bmatrix}$$

$$\theta = \frac{1}{50}\begin{bmatrix} 55 & -15 \\ -15 & 5 \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 \end{bmatrix}\begin{bmatrix} 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}$$

$$= \begin{bmatrix} 1.1 & -0.3 \\ -0.3 & 0.1 \end{bmatrix}\begin{bmatrix} 20 \\ 70 \end{bmatrix}$$

$$\theta = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$\theta_0 = 1, \theta_1 = 1$

Gradient Descent	Closed form
Iterative	non-iterative
α-needed	α- not needed
Works well when n-large (no. of records	Slow when n-large
Support increment learn	$O(n^3)$

Linear Basis Functions

Polynomial basis function

$$\boxed{\phi_j(x) = x^j}$$

Gaussian basis function

$$\phi_j(x) = \exp\left\{-\frac{(x-\mu_j)^2}{2s^2}\right\}$$

μ = mean, s = std. dev.

Sigmoidal basis function

$$\phi_j x = \sigma\left(\frac{x-\mu_j}{s}\right)$$

$$\sigma(a) = \frac{1}{1+e^{-a}}$$

Regularization

Ridge regularization (L2)

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{d}\theta_j^2$$

Lasso regularization (L1)

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{d}|\theta_j|$$

Elastic net :

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \frac{\gamma\lambda}{2}\sum_{j=1}^{d}\theta_j^2 + \frac{(1-\gamma)}{\gamma}\times\sum_{j=1}^{d}|\theta_j|$$

If $\gamma = 0 \Rightarrow$ Lasso regression

$\gamma = 1 \Rightarrow$ Ridge regression

$\gamma \Rightarrow$ Mixration to control the regularization

Gradient of regularization :

Ridge :

$$\theta_j^{new} = \theta_j^{old} - \alpha\frac{1}{n}\left[\sum_{i=1}^{n}(h_\theta(x^{(i)}) - y^{(i)})\,x_j^{(i)} + \lambda\theta_j\right]$$

Lasso :

$$\theta_j^{new} = \theta_j^{old} - \alpha\frac{1}{n}\left[\sum_{i=1}^{n}(h_\theta(x^{(i)}) - y^{(i)})^2\,x_j^{(i)} + \lambda \cdot \text{Sign}(\theta_j)\right]$$

Sign means if $\theta_j < 0$ then -1

$\theta_j > 0$ then 1

Else 0 (ie $\theta_j = 0$)