

PCA Principal Component Analysis

Steps
Date _____
DELTA Pg No. _____

ex	x_1	x_2	Target	New Feature
	4	11		-4.30535
	8	4		3.7361
	13	5		5.6922
	7	14		-5.1288
Sum =	32	34		

Step 1 mean calculation

$$\bar{x}_1 = \frac{32}{4} = 8, \quad \bar{x}_2 = \frac{34}{4} = 8.5$$

Step 2 covariance matrix

$$\begin{matrix} & \bar{x}_1 & \bar{x}_2 \\ \bar{x}_1 & \text{cov}(x_1, x_1) & \text{cov}(x_1, x_2) \\ \bar{x}_2 & \text{cov}(x_2, x_1) & \text{cov}(x_2, x_2) \end{matrix}$$

$$\begin{aligned} \text{cov}(x_1, x_1) &= \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x}_1)^2 \\ &= \frac{(4-8)^2 + (8-8)^2 + (13-8)^2 + (7-8)^2}{4-1} \\ &= \frac{16+0+25+1}{3} = 14 \end{aligned}$$

$$\text{cov}(x_2, x_1) = \text{cov}(x_1, x_2) = -11$$

$$\begin{aligned} \text{cov}(x_2, x_2) &= \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x}_2)^2 \\ &= \frac{(4-8)(11-8.5) + (8-8)(4-8.5) + (13-8)(5-8.5) + (7-8)(14-8.5)}{4-1} \\ &= 23 \end{aligned}$$

$$\begin{aligned} \text{cov}(x_1, x_2) &= \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x}_1)(x_i - \bar{x}_2) \\ &= \frac{(4-8)(11-8.5) + (8-8)(4-8.5) + (13-8)(5-8.5) + (7-8)(14-8.5)}{4-1} \\ &= 23 \end{aligned}$$

Step 3: Eigen vector and eigen values

$$(S - \lambda I) = 0$$

$$\left[\begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right] = 0$$

$$\begin{vmatrix} 14-\lambda & -11 \\ -11 & 23-\lambda \end{vmatrix} = 0$$

$$(14-\lambda)(23-\lambda) - 121 = 0$$

$$\lambda_1 = 30.38, \lambda_2 = 6.61$$

↓
PC1

↓
PC2

Step 4 for each eigen value λ we have corresponding eigen vector $U = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}_{n \times 1}$ such that $(S - \lambda I)U = 0$

$$e = \frac{U}{\|U\|}$$

Date _____
DELTA Pg No. _____

$$(S - \lambda I)U = \begin{bmatrix} 14 - \lambda & -11 \\ -11 & 23 - \lambda \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = 0$$

$$\text{for } \lambda_1 = 30.38 \quad e_1 = \begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix} \quad e_2 = \begin{bmatrix} 0.8303 \\ 0.5574 \end{bmatrix}$$

for $\lambda_1 = 30.38$

$$\begin{bmatrix} 14 - 30.38 & -11 \\ -11 & 23 - 30.38 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = 0$$

$$\begin{bmatrix} -16.38 & -11 \\ -11 & -7.38 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = 0$$

$$\begin{aligned} -16.38U_1 - 11U_2 &= 0 \\ -11U_1 - 7.38U_2 &= 0 \\ U_1 = 0.5574, \quad U_2 &= -0.8303 \end{aligned}$$

$$(14 - \lambda_1)U_1 = 11U_2$$

$$\frac{U_1}{11} = \frac{U_2}{14 - \lambda_1} = t$$

$$U_1 = 11t, \quad U_2 = (14 - \lambda_1)t$$

$$U_1 = \begin{bmatrix} 11 \\ 14 - \lambda_1 \end{bmatrix} \text{ eigen vector}$$

$$11U_1 = (23 - \lambda)U_2 = 0$$

$$\frac{U_1}{23 - \lambda} = \frac{U_2}{11} = t$$

$$U_1 = (23 - \lambda)t, \quad U_2 = 11t$$

$$U_1 = \begin{bmatrix} 23 - \lambda \\ 11 \end{bmatrix}$$

To find a unit eigen vector we compute length

$$\begin{aligned} \|U_1\| &= \sqrt{11^2 + (14 - \lambda)^2} \\ &= \sqrt{11^2 + (14 - 30.38)^2} \\ &= 19.73079 \end{aligned}$$

unit eigen vector for $\lambda_1 = 30.3849$

$$e_1 = \frac{U_1}{\|U_1\|} = \frac{1}{19.73} \begin{bmatrix} 11 \\ 14 - 30.38 \end{bmatrix} = \begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix}$$

$$\begin{aligned} \|U_2\|^2 &= \sqrt{(23 - 6.61)^2 + 11^2} \\ &= 19.7391 \end{aligned}$$

$$e_2 = \frac{U_2}{\|U_2\|} = \frac{1}{19.7391} \begin{bmatrix} 23 - 6.61 \\ 11 \end{bmatrix}$$

$$e_2 = \begin{bmatrix} 0.8303 \\ 0.5574 \end{bmatrix}$$

Step 5:

Compute First principal component.

$$PC_1 = e_1^T \begin{bmatrix} x_{1k} - \bar{x}_1 \\ x_{2k} - \bar{x}_2 \end{bmatrix}$$

for e_1

$$PC_{11} = [0.5574 \quad -0.8303] \begin{bmatrix} 4-8 \\ 11-8.5 \end{bmatrix} = -4.305$$

Target feature

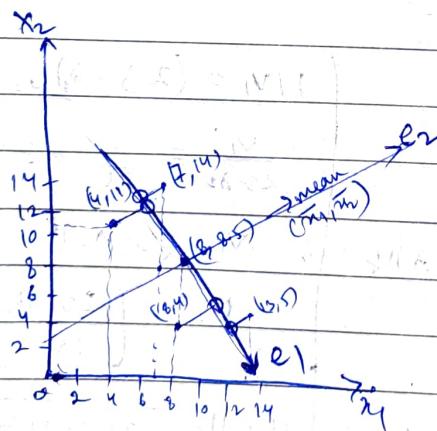
$$PC_{12} = [0.5574 \quad -0.8303] \begin{bmatrix} 8-8 \\ 4-8.5 \end{bmatrix} = 3.73635$$

$$P_{13} = [0.5574 \quad -0.8303] \begin{bmatrix} 13-8 \\ 5-8.5 \end{bmatrix} = 8.69305$$

$$P_{14} = [0.5574 \quad -0.8303] \begin{bmatrix} 7-8 \\ 14-8.5 \end{bmatrix} = -5.12405$$

~~Similarly for 2, 3, 4~~

~~$PC_{21}, P_{22}, P_{23}, P_{24}, \dots$~~



LDA - Linear Discriminant Analysis

LDA is supervised, considers class label and finds components which separates the classes most.

Delta Pg No.

LDA does not change the location of points but tries to maximize the separation of classes.

Fisher Discriminant Ratio - $J(w) = \frac{(\mu_1 - \mu_2)^T}{S_w^{-1} + S_B^{-1}} \rightarrow \text{maximize}$

$$S_w^{-1} + S_B^{-1} \rightarrow \text{minimize}$$

$S_w \rightarrow$ (Inter-class scatter matrix) or scatter matrix within class

$S_B \rightarrow$ (Between-class scatter matrix)

$$J(w) = \frac{w^T S_B w}{w^T S_w w} \Rightarrow \frac{d J(w)}{d w} = 0 \Rightarrow S_w^{-1} S_B w - J_w = 0$$

Ex:- $C_1 \Rightarrow X_1 = (n_1, x_1) = \{(4,1), (2,4), (2,3), (3,6), (4,4)\}$

$C_2 \Rightarrow X_2 = (n_2, x_2) = \{(9,10), (6,8), (9,5), (8,7), (10,8)\}$

Step 1 - Find mean:- $\mu_1 = \left[\frac{4+2+2+3+4}{5}, \frac{1+4+3+6+4}{5} \right] = [3, 3.6]$

$$\mu_2 = \left[\frac{9+6+9+8+10}{5}, \frac{10+8+5+7+8}{5} \right] = [8.4, 7.6]$$

Step 2: Find scatter matrix within class

$$S_1 = \sum_{n \in w_1} (n - \mu_1)(n - \mu_1)^T$$

$$(n - \mu_1) = \begin{bmatrix} (4-3) & (2-3) & (2-3) & (3-3) & (4-3) \\ (1-3.6) & (4-3.6) & (3-3.6) & (6-3.6) & (4-3.6) \end{bmatrix} = \begin{bmatrix} 1 & -1 & -1 & 0 & 1 \\ -2.6 & 0.4 & 0.6 & 2.4 & 0.4 \end{bmatrix}$$

$$S_1 = \begin{bmatrix} 1 & -1 & -1 & 0 & 1 \\ -2.6 & 0.4 & 0.6 & 2.4 & 0.4 \end{bmatrix} \begin{bmatrix} 1 & -2.6 \\ -1 & 0.4 \\ 0 & 0.24 \\ 1 & 0.4 \end{bmatrix}$$

$$= \begin{bmatrix} 1+1+1+0+1 & -2.6 + (-0.4) + 0.6 + 0 + 0.4 \\ 2.6 + 0.4 + 0.6 + 0.4 + 0 & (-2.6)^2 + (0.4)^2 + (0.6)^2 + (2.4)^2 + (0.4)^2 \end{bmatrix} = \begin{bmatrix} 4 & -2 \\ -2 & 13.2 \end{bmatrix}$$

(normalize) $\frac{S_1}{5} = \begin{bmatrix} 0.8 & -0.4 \\ -0.4 & 2.64 \end{bmatrix}$

Similarly $S_2 = \sum [X_2 - \mu_2] [X_2 - \mu_2]^T$

$$[X_2 - \mu_2] = \begin{bmatrix} 9-8.4 & 6-8.4 & 9-8.4 & 8-8.4 & 10-8.4 \\ 10-7.6 & 8-7.6 & 8-7.6 & 7-7.6 & 8-7.6 \end{bmatrix} \rightarrow \begin{bmatrix} 0.6 & -2.4 & 0.6 & -0.4 & 1.6 \\ 2.4 & 0.4 & -2.6 & -0.6 & 0.4 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 0.6 & -2.4 & 0.6 & -0.4 & 1.6 \\ 2.4 & 0.4 & -2.6 & -0.6 & 0.4 \end{bmatrix} \begin{bmatrix} 0.6 & 2.4 \\ -2.4 & 0.4 \\ 0.6 & -2.6 \\ -0.4 & -0.6 \\ 1.6 & 0.4 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 9.2 & -0.2 \\ -0.2 & 13.2 \end{bmatrix} \xrightarrow{\text{(normalize)}} S_2 = \begin{bmatrix} 1.84 & -0.04 \\ -0.04 & 2.64 \end{bmatrix}$$

$$\begin{aligned} S_W &= S_1 + S_2 = \begin{bmatrix} 0.8 & -0.4 \\ -0.4 & 2.64 \end{bmatrix} + \begin{bmatrix} 1.84 & -0.04 \\ -0.04 & 2.64 \end{bmatrix} \\ &= \begin{bmatrix} 2.64 & -0.44 \\ -0.44 & 5.28 \end{bmatrix} \end{aligned}$$

Step 3: Now calculate Between class Scatter matrix (S_B)

$$S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$$

$$= \begin{bmatrix} (3 - 8.4) \\ (3.6 - 7.6) \end{bmatrix}_{2 \times 1} \begin{bmatrix} (3 - 8.4) & (3.6 - 7.6) \end{bmatrix}_{1 \times 2}$$

$$= \begin{bmatrix} -5.4 \\ -4 \end{bmatrix} [-5.4 \quad -4] = \begin{bmatrix} (5.4)^2 & 5.4 \times 4 \\ 5.4 \times 4 & 4^2 \end{bmatrix}$$

$$S_B = \begin{bmatrix} 29.16 & 21.6 \\ 21.6 & 16 \end{bmatrix}_{2 \times 2}$$

Step 4 :- Find LDA projection Vector

$$S_w^{-1} S_B w = \lambda w$$

$$(S_w^{-1} S_B - \lambda I) = 0$$

use calculator

$$S_w^{-1} S_B = \begin{bmatrix} 2.64 & -0.44 \\ -0.44 & 8.28 \end{bmatrix}^{-1} = \begin{bmatrix} 29.16 & 21.6 \\ 21.6 & 16 \end{bmatrix}$$

$$\lambda I = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}$$

$$= \begin{bmatrix} 11.89 & 8.81 \\ 5.08 & 3.76 \end{bmatrix}$$

$$S_w^{-1} S_B - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} = \begin{bmatrix} 11.89 - \lambda & 8.81 \\ 5.08 & 3.76 - \lambda \end{bmatrix} = 0$$

$$\Rightarrow (11.89 - \lambda)(3.76 - \lambda) - 8.81 \times 5.08$$

$$\Rightarrow 11.89 \times 3.76 - (11.89 + 3.76)\lambda + \cancel{\lambda^2} - 8.81 \times 5.08$$

$$\Rightarrow \lambda^2 - 15.65\lambda - 0.04 = 0$$

$$\Rightarrow \lambda = 15.65$$

eigen vector $\Rightarrow S_w^{-1} [\mu_1 - \mu_2]$

$$\begin{bmatrix} 2.64 & -0.44 \\ -0.44 & 8.28 \end{bmatrix}^{-1} \begin{bmatrix} -5.4 \\ -4 \end{bmatrix} = \begin{bmatrix} -5.4 \\ -4 \end{bmatrix}$$

$$= \begin{bmatrix} -2.2016 \\ -0.9608 \end{bmatrix}$$

Eigen vector for $\lambda = \lambda_1 (15.65)$

$$\begin{bmatrix} 11.89 - \lambda & 8.81 \\ 5.08 & 3.76 - \lambda \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 0$$

$$(11.89 - \lambda)v_1 + 8.81v_2 = 0 \Rightarrow \frac{v_1}{-8.81} = \frac{v_2}{11.89 - \lambda} = t$$

$$5.08v_1 + (3.76 - \lambda)v_2 = 0 \Rightarrow \frac{v_1}{5.08} = \frac{v_2}{3.76 - \lambda} = \frac{v_2}{-5.08} = t$$

$$U_1 = \begin{bmatrix} -8.81 \\ 11.89 - 15.65 \end{bmatrix}$$

$$U_2 = \begin{bmatrix} 3.16 - 15.65 \\ -5.08 \end{bmatrix}$$

$$U_1 = \begin{bmatrix} -8.81 \\ -3.76 \end{bmatrix}$$

$$U_{22} = \begin{bmatrix} -12.49 \\ -5.08 \end{bmatrix}$$

$$\|U_1\| = \sqrt{(-8.81)^2 + (-3.76)^2} \\ = 9.5788$$

$$\|U_2\| = \sqrt{(-12.49)^2 + (-5.08)^2} \\ = 13.4835$$

$$e_1 = \frac{U_1}{\|U_1\|} = \begin{bmatrix} -8.81/9.5788 \\ -3.76/9.5788 \end{bmatrix}$$

$$e_2 = \frac{U_2}{\|U_2\|} = \begin{bmatrix} -12.49/13.4835 \\ -5.08/13.4835 \end{bmatrix}$$

$$e_1 = \begin{bmatrix} 0.91 \\ 0.39 \end{bmatrix}$$

$$e_2 = \begin{bmatrix} 0.9263 \\ 0.3787 \end{bmatrix}$$

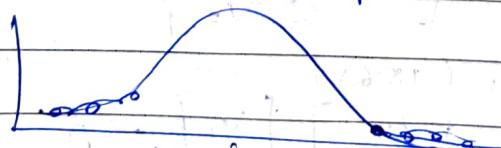
Here in LDA, No, need to calculate PC unlike in PCA

[t-Distributed Stochastic Neighbour Embedding]

t-SNE :- for high dimensional data to 2d It is applied in those area where we can't use PCA

$\frac{\text{Score}}{\text{Sum of all score}}$ } \Rightarrow The more bigger the ratio, the more it is closer to the point

t-distribution, the tail is a bit longer than the normal distributed, And the points also lie on the trail



Step 1 Check whether x_j point is neighbour of x_i or not for this we use:-

$$P_{j|i} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2 / 2\sigma_i^2)}$$

conditional probability, σ_i^2 is the variance of the Gaussian that is centered on x_i datapoint

Step 2: For low-dimensional counterparts y_i and y_j and of high-dimensional datapoints m_i and m_j it is possible to compute a similar conditional probability, which we denote by $q_{j|i}$

$$q_{j|i} = \frac{\exp(-\|y_i - y_j\|^2)}{\sum_{k \neq i} \exp(-\|y_i - y_k\|^2)}$$

(KL divergence)

According to Kullback Leibler divergence method

for discrete probability distribution P and Q defined on sample space X
 relative entropy from Q to P is defined as

$$(loss) \rightarrow D_{KL}(P||Q) = \sum_{x \in X} P(x) \log \left(\frac{P(x)}{Q(x)} \right) \quad q_{j|i}$$

x	0	1	2
distribution $P(x)$	$\frac{9}{25}$	$\frac{12}{25}$	$\frac{4}{25}$
" $Q(x)$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$

$$\begin{aligned} D_{KL}(P||Q) &= \sum_{x \in X} P(x) \ln \left(\frac{P(x)}{Q(x)} \right) \\ &= \frac{9}{25} \ln \left(\frac{9/25}{1/3} \right) + \frac{12}{25} \ln \left(\frac{12/25}{1/3} \right) + \frac{4}{25} \ln \left(\frac{4/25}{1/3} \right) \\ &= 0.08529 \end{aligned}$$

if $P_{j|i}$ is more which means P_j is the neighbour of P_i

loss needs to be less for that $Q_{j|i}$ should be maximum.

$P_{j|i}$ minimum \Rightarrow Therefore log should be minimum

Softmax Regression:-

$$L = \frac{1}{m} \sum_{i=1}^m y_i \log \hat{y}_i + (1-y_i) \log (1-\hat{y}_i)$$

Date _____ Pg No. _____ DELTA _____

$$\frac{\partial L}{\partial w} = -\frac{1}{m} (y - \hat{y}) X$$

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \quad \hat{Y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_m \end{bmatrix}$$

$$w_{new} = w_{old} + \eta (y_i - \hat{y}_i) X$$

$$w_{new} = w_{old} - \eta (y - \hat{y}) X$$

$$\text{Sigmoid func} = \sigma(z) = \frac{1}{1+e^{-z}}$$

Softmax Regression:- is when for k nos of classes

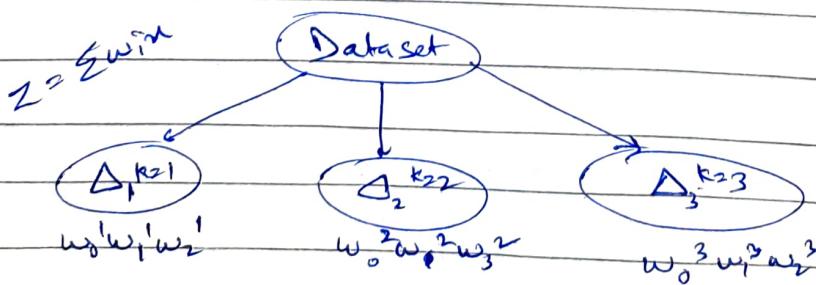
Logistic Regression:- for binary class classification

$$\text{Softmax function} = \sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}}$$

One-hot encoding \rightarrow whenever $y_{k=1} \Rightarrow Y=1$
 put 1 in $y_{k=1}$ and others are 0

x_1	x_2	y	$y_{k=1}$	$y_{k=2}$	$y_{k=3}$
8	7	1	1	0	0
5	3	1	1	0	0
3	4	0	0	1	0
1	2	1	1	0	0
6	8	0	0	0	1

when $y_{k=2}=1$ when $y=2$
 when $y_{k=3}=1$ when $y=3$



$$\sigma(z)_{k=1} = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}} \approx 0.40 \text{ for } k=3$$

$$\because z = \sum_{i=1}^n w_i x_i$$

$$\sigma(z)_{k=2} = \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}} = 0.30$$

Date _____
DELTA Pg No. _____

$$\sigma(z_1) + \sigma(z_2) + \sigma(z_3) = 1$$

$$L = -\frac{1}{m} \sum_{i=1}^n \sum_{k=1}^K y_k^{(i)} \log \hat{y}_k^{(i)}$$

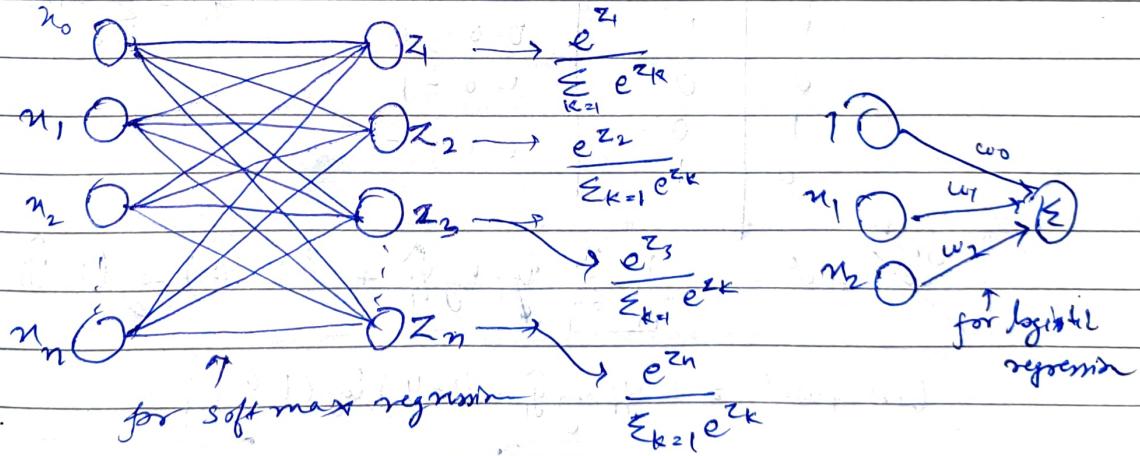
$$L = -\frac{1}{m} \left[\underbrace{\sum_{i=1}^n y_1^{(i)} \log \hat{y}_1^{(i)} + y_2^{(i)} \log \hat{y}_2^{(i)}}_0 + y_3^{(i)} \log \hat{y}_3^{(i)} \right]$$

$$+ \underbrace{\left\{ y_1^{(2)} \log \hat{y}_1^{(2)} + y_2^{(2)} \log \hat{y}_2^{(2)} + y_3^{(2)} \log \hat{y}_3^{(2)} \right\}}_0 + \underbrace{\left\{ y_1^{(3)} \log \hat{y}_1^{(3)} + y_2^{(3)} \log \hat{y}_2^{(3)} + y_3^{(3)} \log \hat{y}_3^{(3)} \right\}}_0$$

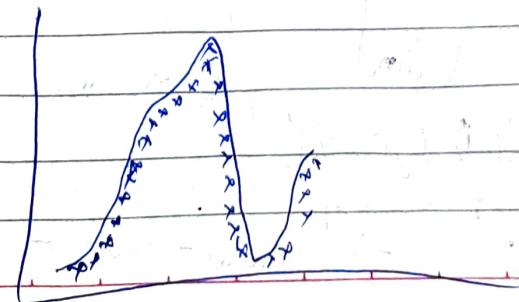
$$L = \frac{1}{m} (y_1^{(1)} \log \hat{y}_1^{(1)} + y_2^{(2)} \log \hat{y}_2^{(2)} + y_3^{(3)} \log \hat{y}_3^{(3)})$$

$y_k^{(i)}$ means \rightarrow i^{th} row, y_k column

x_1	x_2	x_3	y_1	y_2	y_3
x_{11}	x_{12}	1	1	0	0
x_{21}	x_{22}	2	0	1	0
x_{31}	x_{32}	3	0	0	1



⇒ for non-linear data polynomial logistic reg is used, but it does not give best results and efficiency is less.



Logistic Regression :-

$$Ax + By + c = 0$$

$$By = -Ax - c$$

$$y = \frac{A}{B}x + \frac{-c}{B}$$

for multiple inputs

$$Ax_1 + Bx_2 + Cx_3 + D = 0$$

$$Ax_1 + Bx_2 + Cx_3 + Dx_4 + Ex_5 + F = 0$$

$$w_{1m_1} + w_{2m_2} + w_{3m_3} + w_{4m_4} + w_{5m_5} + w_0 = 0$$

$$y = w_0 + w_1x_1 + w_2x_2 \quad] \text{ for 2 points}$$

perception

$$y = \sum_{i=0}^m w_i x_i \quad \text{for } i \geq 0 \quad w_0 = b, x_0 = 1$$

$$y = b + w_1x_1 + w_2x_2$$

$$w_{\text{new}} = w_{\text{old}} + \eta (y_i - \hat{y}_i) x_i, \quad w_{\text{new}} = w_{\text{old}} + \eta \frac{dL}{dw}$$

Actual	Predicted
1	1
0	0
1	0
0	1

$$\hat{y}_i = \sum_{i=0}^m w_i x_i$$

$$1-1=0$$

$$0-0=0$$

$$1-0=1$$

$$0-1=-1$$

$$\text{Cost function} = -\frac{1}{n} \sum_{i=1}^n y_i \log \hat{y}_i + (1-y_i) \log (1-\hat{y}_i)$$

$$\text{Loss function} = y_i \log \hat{y}_i + (1-y_i) \log (1-\hat{y}_i)$$

$$\text{Sigmoid function}, \sigma(z) = \frac{1}{1+e^{-z}}$$

$$\frac{d\sigma(z)}{dz} = \sigma'(z) = \sigma(z)(1-\sigma(z))$$

$$\hat{y} = \sigma(x^T w) \Rightarrow$$

$$\hat{y} = \sigma(z)$$

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_m \end{bmatrix} = \sigma \left(\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix} \times \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \right)$$

$$L = -\frac{1}{m} (y \log \hat{y} + (1-\hat{y}) \log (1-\hat{y}))$$

$$\frac{\partial L}{\partial w} = -\frac{1}{m} (y - \hat{y}) x$$

$$\Rightarrow w_{\text{new}} = w_{\text{old}} - \eta \frac{(y - \hat{y}) x}{m}$$

Classification

		predicted	
		TN	FP
		FN	TP
Actual			

D → Negative
 1 → positive
 for predicted
 T if actual = pred
 F if actual ≠ pred

$$(A) \text{ Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{\text{no. of correctly classified}}{\text{Total no. of prediction}}$$

$$(P) \text{ Precision} = \frac{TP}{TP + FP} = \text{what proportion of predicted positive is True positive}$$

$$(R) \text{ Recall} = \frac{TP}{TP + FN} = \text{what proportion of Actual positive is True positive}$$

$$F1 \text{ score} = \frac{2 PR}{P+R}$$

High P
Low R

Low P
High R

Recall = Precision when, $FP = FN$

$$\frac{TP}{TP + FP} = \frac{TP}{TP + FN}$$

$$FP = FN$$

		predicted	
		0	1
Actual	Not Spam	700	30
	Spam	170	100

A

		predicted	
		0	1
Actual	Not Spam	700	10
	Spam	190	100

$$\text{Accuracy} = \frac{700 + 100}{700 + 30 + 170 + 100} = \frac{800}{900} = 80\%$$

$$\text{Precision} = \frac{100}{100 + 30} = 76.9\% \quad P = \frac{100}{100 + 100} = 50\%$$

$$\text{Recall} = \frac{100}{100 + 170} = 37.03\% \quad \frac{100}{100 + 190} = 34.48\%$$

R > P For rare cancer data modelling

P > R For YouTube recommendation, false-negative is a concern

Actual		Dog	Cat	Rabbit	Σ predicted	Accuracy
Predicted		25	5	10	40	$\frac{25 + 30 + 20}{25 + 5 + 10 + 0 + 30 + 4 + 4 + 10 + 20} = \frac{75}{108} = 0.694$
Dog		25	5	10	40	
Cat		0	30	4	34	$\text{Precision}_{\text{Dog}} = \frac{25}{29} = 0.862, P_{\text{Cat}} = \frac{30}{45} = 0.66, P_{\text{Rabbit}} = \frac{4}{34} = 0.118$
Rabbit		4	10	20	34	
Σ Actual		29	45	34	108	$\text{Macro Precision} = \frac{0.862 + 0.66 + 0.118}{3} = 0.588$

$$\text{Weighted Precision} = \frac{40 \times 0.862 + 34 \times 0.66 + 34 \times 0.118}{108} = 0.588$$

$$\text{Recall}_{\text{Dog}} = \frac{25}{40} = 0.625, R_{\text{Cat}} = \frac{30}{34} = 0.882, R_{\text{Rabbit}} = \frac{20}{34} = 0.588 \quad = 0.714$$

$$\text{Macro Recall} = 0.625 + 0.882 + 0.588 = 0.698$$

$$\text{Weighted Recall} = \frac{25 \times 0.625 + 29 \times 0.882 + 45 \times 0.588}{108} = 0.7204$$

~~McCulloch Pitts model of a neuron~~

Perceptron

Training Algo:-

Step1: Initialize weights and bias (initially can be 0)
Set learning rate α (0 to 1)

Step2: While stopping condition is false do steps 3 to 7

Step3: For each training pair (x, t) do steps 4-6

Step4 Set activations of input units

$$n_i = x_j \text{ for } i=1 \text{ to } n$$

Step5: Compute the output unit response

$$y_{in} = b + \sum x_i w_i$$

The activation func used is,

$$y = f(y_{in}) = \begin{cases} 1 & \text{if } y_{in} > 0 \\ 0 & \text{if } y_{in} \leq 0 \\ -1 & \text{if } y_{in} < 0 \end{cases}$$

Step6: The weights and bias are updated if the target is not equal to the output response

if $t \neq y$ and value of n_i is not 0

$$w_{new} = w_{old} + \alpha t x_i$$

$$b_{new} = b_{old} + \alpha t$$

else

$$w_{new} = w_{old}, b_{new} = b_{old}$$

Step7: Test for stopping condition :- stopping condition may be weight changes

Note: 1) only weights connecting active input units ($n_i \neq 0$) are updated

2) weights are update only for patterns that don't produce the correct value

Testing algo:-

Step1: The weights to be used here are taken from the training algo

Step2: For each input vector x to be classified do steps 3-4

Step3: Input units activations are set

Step4: Calculate response of output unit

$$y_{in} = \sum x_i w_i, y = f(-y_{in}) = \begin{cases} 1 & \text{if } y_{in} > 0 \\ 0 & \text{if } 0 \leq y_{in} \leq \theta \\ -1 & \text{if } y_{in} < 0 \end{cases}$$

