Page No.:
Date:

① problem statement: realestate company wants to predict house prices based on features like size (in square feet), no. of bedrooms and house age.

Size	Bedrooms	Age	Price ($)
2100	3	30	400,000
1600	2	20	300,000
2400	4	15	500,000
1400	3	40	250,000

X ~~[illegible]~~ : matrix of input features
w : vector w

$$Xw \approx y$$

$$X = \begin{bmatrix} 2100 & 3 & 30 \\ 1600 & 2 & 20 \\ 2400 & 4 & 15 \\ 1400 & 3 & 40 \end{bmatrix}, \quad y = \begin{bmatrix} 400000 \\ 300000 \\ 500000 \\ 250000 \end{bmatrix}$$

weight vector, $w = [w_0 \; w_1 \; w_2 \; w_3]^T$

* Compute Q and R using QR decomposition:

$$Q = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ q_{41} & q_{42} & q_{43} & q_{44} \end{bmatrix}$$

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ 0 & r_{22} & r_{23} & r_{24} \\ 0 & 0 & r_{33} & r_{34} \\ 0 & 0 & 0 & r_{44} \end{bmatrix}$$

* Q (orthogonal matrix)
* R (upper triangular matrix)

* add 1 to another column in X for matrix multiplication

$$X = \begin{bmatrix} 1 & 2100 & 3 & 30 \\ 1 & 1600 & 2 & 20 \\ 1 & 2400 & 4 & 15 \\ 1 & 1400 & 3 & 40 \end{bmatrix}$$

$$X = QR$$

$$y = X\beta = QR\beta$$

$$Q = \begin{bmatrix} -0.5, & 0.284, & 0.289, & 0.765 \\ -0.5, & -0.3471, & 0.655, & -0.446 \\ -0.5, & 0.602, & -0.334, & -0.446 \\ -0.5, & -0.599, & -0.611, & 0.127 \end{bmatrix}$$

$$R = \begin{bmatrix} -2 & -3.75 & -6 & -52.5 \\ 0 & 7.92 & 1.009 & -12.4 \\ 0 & 0 & -.98 & -7.66 \\ 0 & 0 & 0 & 12.4 \end{bmatrix}$$

Page No.:
Date:

(2)

Problem statement: The company wants to optimize product deliveries between warehouses and stores to minimize costs.

W1	1	1	0	100
W2	0	2	1	150
W3	1	0	1	200

matrix representation:

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 100 \\ 150 \\ 200 \end{bmatrix}$$

x_1, x_2 and x_3 are the amounts delivered to store1, store 2 and store3.

LU Decomposition to decompose the coefficient matrix into lower and upper triangular matrices.

$$Ax = b$$

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 2 & 1 \\ 1 & 0 & 1 \end{bmatrix}, \quad X = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix}, \quad b = \begin{bmatrix} 100 \\ 150 \\ 200 \end{bmatrix}$$

Decompose A into L and U matrices:

$$A = LU$$

L is a lower triangular matrix:

$$L = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix}$$

U is an upper triangular matrix:

$$U = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 2 & 1 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ l_{21}u_{11} & l_{21}u_{12}+u_{22} & l_{21}u_{13}+u_{23} \\ l_{31}u_{11} & l_{31}u_{12}+l_{32}u_{22} & l_{31}u_{13}+l_{32}u_{23}+u_{33} \end{bmatrix}$$

* $u_{11} = 1, \quad u_{12} = 1, \quad u_{13} = 0$

* $l_{21}u_{11} = 0 \Rightarrow l_{21} = 0$
* $l_{21}u_{12} + u_{22} = 2 \Rightarrow u_{22} = 0$

* $l_{31}u_{11} = 1 \Rightarrow l_{31} = 1$

* $l_{21}u_{13} + u_{23} = 1 \Rightarrow u_{23} = 1$

* $l_{31}u_{12} + l_{32}u_{22} = 0$

$l_{32} = 0$

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

$LY = B$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 100 \\ 150 \\ 200 \end{bmatrix}$$

- $y_1 = 100$
- $y_2 = 150$
- $y_3 = 200 - y_1 = 100$

$$y = \begin{bmatrix} 100 \\ 150 \\ 100 \end{bmatrix}$$

* $UX = y$ for x

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 100 \\ 150 \\ 100 \end{bmatrix}$$

* $x_3 = 100$
* $2x_2 + x_3 = 150 \Rightarrow x_2 = \frac{150 - 100}{2} = 25$
* $x_1 + x_2 = 100 \Rightarrow x_1 = 100 - 25 = 75$

$$X = \begin{bmatrix} 75 \\ 25 \\ 100 \end{bmatrix}$$

optimal product distribution to meet the demand:

- $x_1 = 75$ units to Store 1.
- $x_2 = 25$ units to Store 2.
- $x_3 = 100$ units to Store 3.

—x—————x—

(3)

$$(M)\ \text{Image Matrix} = \begin{bmatrix} 100 & 102 & 105 & 107 & 109 \\ 102 & 104 & 107 & 110 & 112 \\ 105 & 107 & 110 & 113 & 115 \\ 107 & 110 & 113 & 115 & 118 \\ 109 & 112 & 115 & 118 & 120 \end{bmatrix}$$

* SVD(Singular Value Decomposition)

M can be decomposed as:

$$M = U . \Sigma . V^T$$

- U is an orthogonal matrix of left singular vectors.
- Σ is a diagonal matrix containing singular values.
- V^T is the transpose of an orthogonal matrix of right singular vectors.

* • Compute matrix U:

- MM^T compute
- Eigen Values = [302079.56, 0.764, 0.518, 0.158, 0.004]
- Corresponding Eigenvectors (forming U) =

$$U = \begin{bmatrix} 0.4258 & -0.2349 & -0.7963 & 0.3014 & 0.1964 \\ 0.4356 & 0.4809 & 0.1098 & -0.3758 & 0.6525 \\ 0.4478 & 0.4270 & +0.1595 & -0.2684 & -0.7287 \\ 0.4584 & -0.7256 & 0.2836 & -0.4231 & -0.0625 \\ 0.4673 & 0.0684 & 0.4979 & 0.7268 & -0.0275 \end{bmatrix}$$

- **Compute Σ:**

- **Eigenvalues of MM^T:**

$$\lambda = [302079.56, 0.764, 0.518, 0.158, 0.004]$$

- **Take square roots of Eigenvalues:** (Singular values)

$$\sigma = [549.62, 0.874, 0.72, 0.398, 0.065]$$

- **Construct Σ:**

$$\Sigma = \begin{bmatrix} 549.62 & 0 & 0 & 0 & 0 \\ 0 & 0.874 & 0 & 0 & 0 \\ 0 & 0 & 0.72 & 0 & 0 \\ 0 & 0 & 0 & 0.398 & 0 \\ 0 & 0 & 0 & 0 & 0.065 \end{bmatrix}$$

- **Compute V:**

- Compute M^TM.
- Find Eigenvalues and Eigenvectors of M^TM.
- Sort Eigenvalues and arrange corresponding eigenvectors.
- Eigenvectors form the columns of V.
- Transpose V to Get V^T

$$V^T = \begin{bmatrix} -0.4257 & -0.435 & -0.447 & -0.458 & -0.467 \\ 0.234 & -0.480 & -0.426 & 0.725 & -0.068 \\ 0.796 & -0.109 & 0.159 & -0.283 & -0.497 \\ -0.301 & 0.375 & 0.246 & 0.423 & -0.726 \\ -0.196 & 0.652 & -0.728 & -0.062 & -0.027 \end{bmatrix}$$

Compression:

- we can retain only the largest singular value (549.62) or the first few largest singular value to approximate the image matrix.

- retaining only a few singular values reduces the matrix size while preserving most of the images information.

X ————————— X

(4) Problem statement: A retail company collects data on customer purchases, and they want to reduce the dimensionality of the dataset for better visualization and analysis.

Objective: perform PCA.

Data:

Customer	Product 1	Product 2	Product 3
C1	120	80	50
C2	150	90	60
C3	100	70	45
C4	130	85	55
C5	140	95	65

representation of Data-

$$X = \begin{bmatrix} 120 & 80 & 50 \\ 150 & 90 & 60 \\ 100 & 70 & 45 \\ 130 & 85 & 55 \\ 140 & 95 & 65 \end{bmatrix}$$

Page No.:
Date:

Calculation of Mean of Each column -

Mean Product 1 : $\frac{120+150+100+130+140}{5} = 1$

Mean Product 2 : $\frac{80+90+70+85+95}{5} = 84$

Mean Product 3 : $\frac{50+60+45+55+65}{5} = 55$

Center of the Data:

[X - mean of column]

$$X_{centered} = \begin{bmatrix} 120-128 & 80-84 & 50-55 \\ 150-128 & 90-84 & 60-55 \\ 100-128 & 70-84 & 45-55 \\ 130-128 & 85-84 & 55-55 \\ 140-128 & 95-84 & 65-55 \end{bmatrix}$$

$$= \begin{bmatrix} -8 & -4 & -5 \\ 22 & 6 & 5 \\ -28 & -14 & -10 \\ 2 & 1 & 0 \\ 12 & 11 & 10 \end{bmatrix}$$

Covariance matrix:

$$\Sigma = \frac{1}{n-1} X^T_{centered} X_{centred}$$

$n = 5$, (no. of observations)

1. Calculate the product $X^T_{centered} X_{centered}$

$$X^T_{centered} X_{centered} = \begin{bmatrix} -8 & 22 & -28 & 2 & 12 \\ -4 & 6 & -14 & 1 & 11 \\ -5 & 5 & -10 & 0 & 10 \end{bmatrix} \begin{bmatrix} -8 & -4 & -5 \\ 22 & 6 & 5 \\ -28 & -14 & -10 \\ 2 & 1 & 0 \\ 12 & 11 & 10 \end{bmatrix}$$

$$= \begin{bmatrix} 1524 & 558 & 460 \\ 558 & 186 & 155 \\ 460 & 155 & 130 \end{bmatrix}$$

2. Divide by $n-1 = 5-1 = 4$:

$$\Sigma = \frac{1}{4} \begin{bmatrix} 1524 & 558 & 460 \\ 558 & 186 & 155 \\ 460 & 155 & 130 \end{bmatrix} = \begin{bmatrix} 381 & 139.5 & 115 \\ 139.5 & 46.5 & 38.75 \\ 115 & 38.75 & 32.5 \end{bmatrix}$$

Page No.:
Date:

Finding Eigenvalues and Eigenvectors of the covariance matrix:

$$\det(\Sigma - \lambda I) = 0$$

$$\Sigma - \lambda I = \begin{bmatrix} 381-\lambda & 139.5 & 115 \\ 139.5 & 46.5-\lambda & 38.75 \\ 115 & 38.75 & 32.5-\lambda \end{bmatrix}$$

$$\det(\Sigma - \lambda I) = (381-\lambda)\left[(46.5-\lambda)(32.5-\lambda) - (38.75)^2\right]$$
$$-(139.5)\left[(139.5)(32.5-\lambda) - (115)(38.75)\right]$$
$$+(115)\left[(139.5)(38.75) - (115)(46.5-\lambda)\right]$$

$$\lambda = \underset{(\lambda_1)}{456.13},\ \underset{(\lambda_2)}{3.24},\ \underset{(\lambda_3)}{0.63} \quad \text{(Eigenvalues)}$$

Eigenvectors -

for λ_1:

$$\begin{bmatrix} 381-\lambda_1 & 139.5 & 115 \\ 139.5 & 46.5-\lambda_1 & 38.75 \\ 115 & 38.75 & 32.5-\lambda_1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\lambda_1 : \begin{bmatrix} 0.87 \\ 0.35 \\ 0.34 \end{bmatrix}$$

$$\lambda_2 : \begin{bmatrix} -0.23 \\ 0.91 \\ 0.33 \end{bmatrix}$$

$$\lambda_3 : \begin{bmatrix} -0.43 \\ -0.20 \\ 0.88 \end{bmatrix}$$

- Principal components are λ_1 and λ_2

Transforming of Data:

$$W = \begin{bmatrix} 0.87 & -0.23 \\ 0.35 & 0.91 \\ 0.34 & 0.33 \end{bmatrix}$$

transformed data Y is:

$$Y = X\,\text{centered} \cdot W$$

$$= \begin{bmatrix} -8 & -4 & -5 \\ 22 & 6 & 5 \\ -28 & -14 & -10 \\ 2 & 1 & 0 \\ 12 & 11 & 10 \end{bmatrix} \begin{bmatrix} 0.87 & -0.23 \\ 0.35 & 0.91 \\ 0.34 & 0.33 \end{bmatrix}$$

Page No.:
Date:

Each row of Y represents the data in the new reduced-dimensional space.

x ——— x

Page No.
Date

CS 1

(5) **Problem Statement:** A company wants to analyze customer reviews to determine overall sentiment (+ve or -ve)

Objective: To represent text data as vectors using word embeddings and perform operations to classify the sentiment.

Data:

1. **Review Text:**

- "The product is amazing!" → Positive
- "I hated the experience." – Negative
- "It's okay, not great." → Neutral

2. **Sentiment:**

(positive, Negative or Neutral)

* **Processing the Data**

1. Tokenization: tokenize the text by breaking it into words.
2. removing stop words.
3. Lemmatization (optional)

Preprocessed data:

['The', 'product', 'is', 'amazing']

['I', 'hated', 'the', 'experience']

['9t's', ~~'not'~~ 'okay', 'not', 'great']

Output:

[(['product', 'amazing'], 'positive'),

(['hate', 'experience'], 'negative'),

(['okay', 'great'], 'neutral')]

↳ Conversion of each review into a vector

Output:

[(array([0.1, -0.05, ...]), 'positive'),

(array([-0.2, 0.1, ..., -0.01]), 'negative'),

(array([0.05, 0.02, ..., 0.04]), 'neutral')]

↳ each review has been converted to a 100-dimensional array.