Assignment 1

Pradyoth Singenahalli Prabhu – 02071847
Woodi Raghavendra Varun – 02070206
Bharath Anand – 02044023

1. Data mining concepts

a.

1. In this method of sampling, the number of elements selected is in direct proportion to m times n. The random selection of elements is done without considering the grouping of objects within the data set.

2. In this sampling method, the proportion of each group in the sample is determined solely by the number of elements selected from that group and not by the size of the group in the entire population.

b.

1. Time in terms of AM or PM: **Categorical** – Nominal.
2. ISBN numbers for books: **Numeric** – Continuous.
3. Angles as measured in degrees between and 0 and 360: **Numeric** – Continuous.
4. Brightness as measured by a light meter: **Numeric** – Continuous.
5. Brightness as measured by people's judgments: **Categorical** – Ordinal.
6. Bronze, Silver, and Gold medals as awarded at the Olympics: **Categorical** – Ordinal.
7. Height above sea level: **Numeric** – Continuous.
8. Number of patients in a hospital: **Numeric** – Discrete.

2. Matrices Concepts

a.

Step-1 : $RYA(A^{-1}) = B(A^{-1})$

Step-2 : Using $AA^{-1} = I$

$RYI = B(A^{-1})$

Step-3 : R^{-1} on both sides

$RY(R^{-1}) = BA^{-1}(R^{-1})$

Step-4 : Similarly $R(R^{-1}) = I$

$\boxed{Y = BA^{-1}R^{-1}}$

b.

$$A = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$$

$$AB = \begin{bmatrix} 0 & 2 \\ 0 & 0 \end{bmatrix}$$

$$(AB)C = \begin{bmatrix} 4 & 2 \\ 0 & 0 \end{bmatrix} \longrightarrow (1)$$

$$BC = \begin{bmatrix} 2 & 1 \\ 0 & 0 \end{bmatrix}$$

$$A(BC) = \begin{bmatrix} 4 & 2 \\ 0 & 0 \end{bmatrix} \longrightarrow (2)$$

$\therefore$ From (1) and (2) we can see

$$\boxed{(AB)C = A(BC)}$$

c.

c)
$$A = \begin{bmatrix} 2 & -2 & -4 \\ -1 & 3 & 4 \\ 1 & -2 & -3 \end{bmatrix} \longrightarrow (1)$$

$$A^2 = A \cdot A = \begin{bmatrix} 2 & -2 & -4 \\ -1 & 3 & 4 \\ 1 & -2 & -3 \end{bmatrix} \begin{bmatrix} 2 & -2 & -4 \\ -1 & 3 & 4 \\ 1 & -2 & -3 \end{bmatrix}$$

$$A^2 = \begin{bmatrix} 2 & -2 & -4 \\ -1 & 3 & 4 \\ 1 & -2 & -3 \end{bmatrix} \longrightarrow (2)$$

So (1) = (2)

i.e. $\boxed{A = A^2}$

Hence the matrix A is Idempotent matrix

It is indeed feasible to locate an idempotent matrix whose determinant is non-zero. An identity matrix is a representative example of such a matrix. The identity matrix is a square matrix with 1s present along its main diagonal and 0s in the other elements. The property of being idempotent is derived from the fact that when multiplied by itself, an identity matrix results in itself. The determinant of an identity matrix is calculated as the product of the elements in its diagonal, which is 1. Therefore, an identity matrix is an idempotent matrix with a determinant value of 1, which is not equal to zero.

3.

a.

$$\|A\|_2 = \max_x \sqrt{(Ax)^T \ (Ax)}$$

$$= \max_x \sqrt{(U \Sigma V^T x)^T \ (U \Sigma V^T x)}$$

$$\|A\|_2 = \max_x \sqrt{x^T V \Sigma^T \ U^T U \Sigma V^T x}$$

Here $U^T U = I$

$$\|A\|_2 = \max_x \sqrt{x^T V \Sigma^T \Sigma V^T x}$$

Here x is unit matrix and V is rotation matrix.

So, 'w' is a unit vector produced by Vx.

Similarly, 'w^T' is produced by $V^T x^T$.

$$\|A\|_2 = \max_x \sqrt{w w^T \Sigma^T \Sigma}$$

$$\|A\|_2 = \max_x \sqrt{\Sigma^T \Sigma} \quad \because w w^T = I$$

Let $\sigma_1, \sigma_2 \ldots\ldots \sigma_n$ be the diagonal elements of diagonal matrix 'Σ'

$$\therefore \boxed{\|A\|_2 = \sigma_{max}}$$

σ_{max} is the largest singular value of 'A'.

b.

$$A^T A = (U \Sigma V^T)^T \ (U \Sigma V^T)$$

$$A^T A = U^T \Sigma^T U^T \ U \Sigma V^T$$

Here $U U^T = I$ and $V V^T = I$

$$A^T A = \Sigma^T \Sigma$$

Applying Trance

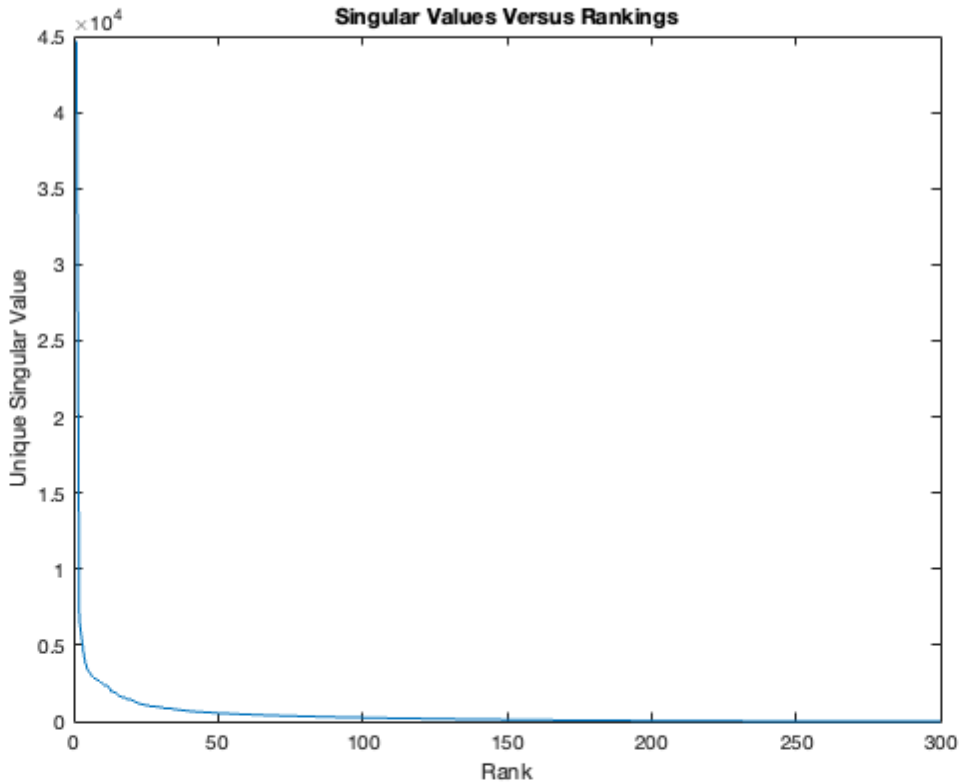
Trance$(\Sigma^T \Sigma)$

$$\text{trance}(A^T A) = \sigma_1^2 + \sigma_2^2 + \sigma_3^2 \ldots\ldots + \sigma_n^2$$

4.
A

```
% Reading the flower.bmp image and converting it into double format and grayscale.
Image = imread('/Users/bharath/Documents/MATLAB/flower.bmp');
Image = double(im2gray(Image));

% Perform SVD on the flower.bmp image
[U, S, V] = svd(Image);

% Extract the top 10 singular values from the flower.bmp image
top_10_singular_values = diag(S(1:10,1:10));

% Plot each unique singular value against its rating.
figure;
plot(1:length(diag(S)), diag(S));
xlabel('Rank');
ylabel('Unique Singular Value');
title('Singular Values Versus Rankings');

% Print the top 10 singular values
fprintf('Top 10 singular values: \n');
disp(top_10_singular_values);
```

```
Top 10 singular values:
   1.0e+04 *

    4.4672
    0.6623
    0.5166
    0.3875
    0.3328
    0.3069
    0.2853
    0.2773
    0.2651
    0.2455
```



Published with MATLAB® R2022a

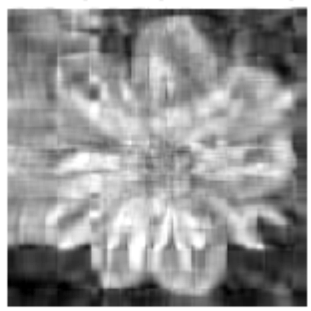
You will notice from the plot that the singular values decrease rapidly, which means that most of the information in the image can be represented by the first few singular values. This is why SVD can be effectively used for image compression. The top 10 singular values contain the most significant information in the image.

4.
B

```
% Read the flower.bmp image and convert it to grayscale and double format
Image = imread('/Users/bharath/Documents/MATLAB/flower.bmp');
Image = double(im2gray(Image));

% Perform SVD on the image
[U, S, V] = svd(Image);

% Reconstruct and display the original image using SVD matrices
Image_reconstructed = U * S * V';
figure;
imshow(uint8(Image_reconstructed));
title('Reconstructed Image using SVD');

% Compress the image using top k singular values and corresponding left/right singular vectors
k_values = [10, 50, 100];
for i = 1:length(k_values)
    k = k_values(i);
    Image_compressed = S;
    Image_compressed(k+1:end,k+1:end) = 0;
    Image2_compressed = U * Image_compressed * V';
    figure;
    imshow(uint8(Image2_compressed));
    title(sprintf('Compressed Image using Top %d Singular Values', k));
end
```

Reconstructed Image using SVD

Compressed Image using Top 10 Singular Values



Compressed Image using Top 50 Singular Values

Compressed Image using Top 100 Singular Value:

As you increase the value of k, you'll find that the compression is less obvious and the rebuilt image resembles the original image more. This is due to the fact that using more singular values enables the image to capture and reconstruct more data. However, employing a lower value of k causes a more significant compression and information loss in the image, which could result in an image quality decrease.

4.
C

```
% Read the flower.bmp image and convert it to grayscale and double format
Image = imread('/Users/bharath/Documents/MATLAB/flower.bmp');
Image = double(im2gray(Image));

% Perform SVD on the image
[U, S, V] = svd(Image);

% Reconstruct and display the original image using SVD matrices
Image_reconstructed = U * S * V';
figure;
imshow(uint8(Image_reconstructed));
title('Reconstructed Image using SVD');

% Compress the image using top k singular values and corresponding left/right singular vectors
k_values = 200;
for i = 1:length(k_values)
    k = k_values(i);
    Image_compressed = S;
    Image_compressed(k+1:end,k+1:end) = 0;
    Image2_compressed = U * Image_compressed * V';
    figure;
    imshow(uint8(Image2_compressed));
    title(sprintf('Compressed Image using Top %d Singular Values', k));
end
```

The dimensions of the image are dependent on the singular value decomposition (SVD) values, which are stored as a diagonal matrix. The size of the matrix is determined by the dimensions of the original image. For example, if k=200, the matrix size would be 200x200.

With k=200. The original image is of size m x n, while the left singular value matrix has a size of m x min(m x n), and the right singular value matrix has a size of min(m x n) x n. As we have taken k=200 the dimensions of the image will be transformed to m x 200 + 200 x 200 + 200 x n, significantly smaller than its original size m x n. The value of k being set to 200 presents a chance to reduce the size of the image, but this may not always be the best option. It's crucial to choose the value of k in a way that strikes a balance between the size of the compressed or reconstructed image for optimal results.