** PES University, Bangalore**

(Established under Karnataka Act No. 16 of 2013)

**MAY 2020: IN SEMESTER ASSESSMENT (ISA) B.TECH. IV SEMESTER**

**UE18MA251- LINEAR ALGEBRA**

MINI PROJECT REPORT

ON

Applications of Linear Algebra

Submitted by

1. Name: Ahan Bose SRN:PES1201800467

Branch & Section : ECE & C Section

PROJECT EVALUATION

( For Official Use Only )

|  |  |  |  |
| --- | --- | --- | --- |
| Sl.No. | Parameter | Max Marks | Marks Awarded |
| 1 | Background & Framing of the problem | 4 |  |
| 2 | Approach and Solution | 4 |  |
| 3 | References | 4 |  |
| 4 | Clarity of the concepts & Creativity | 4 |  |
| 5 | Choice of examples and understanding of the topic | 4 |  |
| 6 | Presentation of the work | 5 |  |
|  | Total | 25 |  |

Name of the Course Instructor : VRINDA KAMATH .K

Signature of the Course Instructor :

Overview:

Linear Algebra finds immense application in the domain of machine learning. Machin learning is an emerging field and modern day applications simply cannot function without artificial intelligence. However with highly powerful modern-day libraries and programming languages like R/Python available for use we tend to forget the mathematics behind the making of these libraries.

Section A of this paper dives deeper into the mathematical explanation of Linear Regression and solves a regression analysis on large dataset of salaries of employees while establishing a correlation between the salary range and years of experience. This analysis is purely based on the use of projection matrices and error minimalization using Linear Algebra only. The same model is them verified using powerful python machine learning libraries.

Section B of this paper explores the use of Singular Value Decomposition in recommendation system and displays the power of matrix factorisation in solving some complex algorithms which we make use of in our daily lives without understanding the calculations behind it.

Section C presents another application of singular value decomposition under the domain of image processing. SVD can be used for dimensionality reduction and hence compress a digital image to occupy lesser memory.

**SECTION A: Regression Analysis using Linear Algebra**

Introduction:

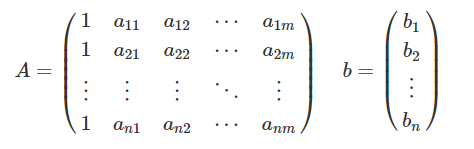
**Linear regression** is a method for modelling the relationship between two scalar values: the input variable x and the output variable y. The goal of regression is to establish a correlation, if one exists between the dependent variable(y) also known as outcome variable and one or more independent variable also called features. Most common approach for regression is usually minimising error using calculus, however we explore the use of linear algebra for the same in this paper. We perform Regression on a dataset of salaries of an employee and the corresponding years of experience and establish a correlation between the two.

Review of Literature:

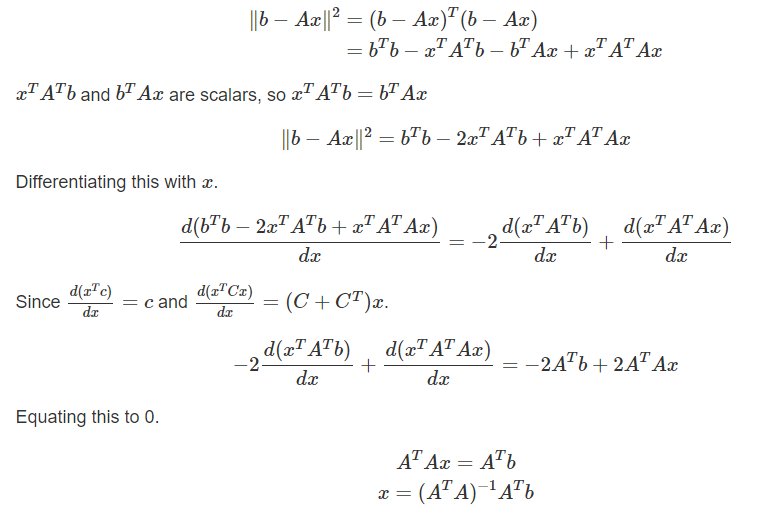
Calculus is the preferred approach to explain linear regression. Here I present the use of calculus to derive an expression.

Assume we have a system Ax=b, if b is not in the column space then gaussian elimination fails and the system as no solution. System is said to be inconsistent. However, as Gilbert Strang himself explains it , in spite of their insolvability inconsistent equations arise all the time and they have to be solved.[1][3.3][Pg 180] The solution is finding an approximate x’ which minimises an average error E in m equations.

Assume n data points and observation pairs (a1,b1),(a2,b2)…..(an,bn) where ai∈Rm and bi∈R. Let Ai∈Rm+1 be the vector obtained by prepending 1 to vector ai. The matrix A has Ai as its ith row and b∈Rn  is the column vector consisting of bi.



We are supposed to find the vector xx such that ∥b−Ax∥2  is minimum. What this means is vector Ax is as close as possible to vector b. Differentiate ∥b−Ax∥2 with x and equate it to zero.



Report on present Investigation:

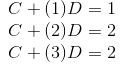
Now we derive the same formula using Linear Algebra.

Assume we are collecting data on the number of machine failures of a robotic arm in a manufacturing plant, per day.we have 3 data points (1,1),(2,2),(3,2). Goal is to fit a linear equation that fits these points, which would map days uniquely to machine failures.

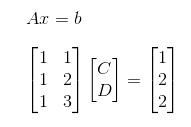
[2]

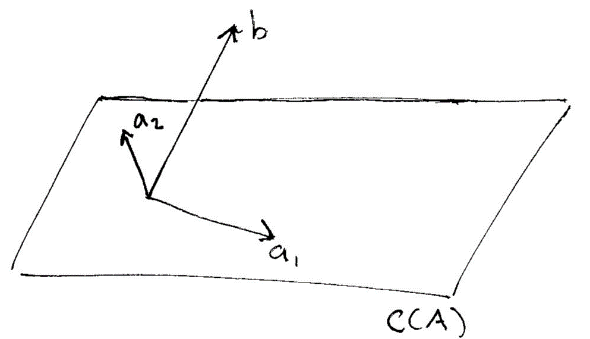
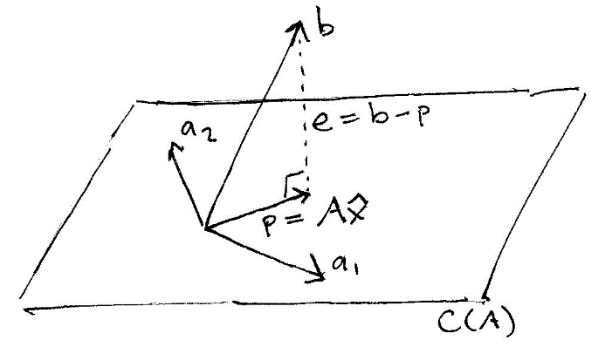
F: days->failures

B=C+Dx



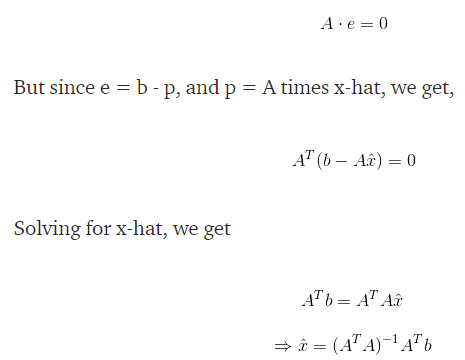
It is obvious the three points do not sit on the same line ,hence for system Ax=b b is not in the column space of A and system is inconsistent.



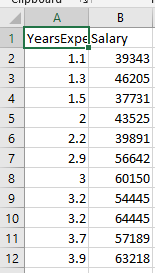
e is perpendicular to A

so we find the approximate x that solves the system lets call it x^(x hat)

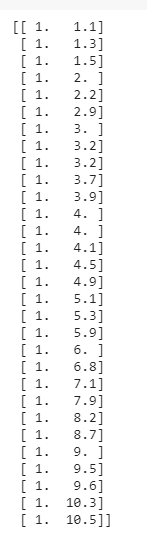


Output/Result:

We have a dataset of salaries vs years of experiences. There are 30 rows. A snippet of the dataset in shown below:



Now we perform linear regression using Linear Algebra using python

A matrix

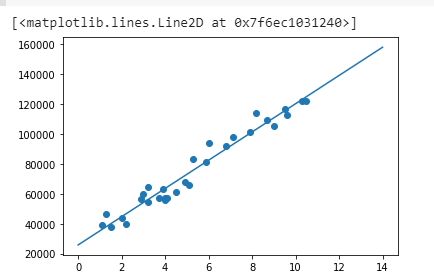
After calculating the slope and intercept using the derived formula: [1] [3.3]



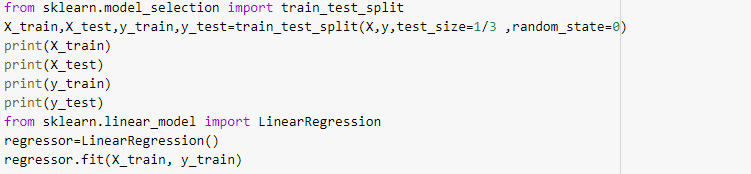




Output graph



TRAININED MODEL USING PYTHON ML LIBRARIES





Hence we can conclude the outputs are the same.

**Summary/Conclusion:**

By observing the graph obtained by using the sklearn library and the the model obtained from using projection matrices we can conclude that the slope and intercept is the same. This is further verified by the presented derivations.

**SECTION B: Movie recommendation system using singular value decomposition**

**Introduction:**

How does Netflix predict which movies we would like to watch? Or How does spotify know our favourity music album?

The answer is collaborative filtering.

What is a recommendation system?

When Amazon suggests to you the other products that people have bought alongside the one that you are buying. This is an example of a recommendation engine.

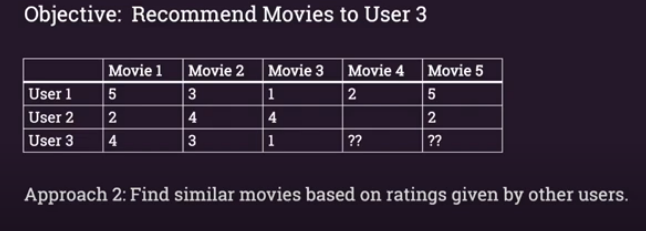
**Report on Current Investigation:**

There are 2 types of recommendation system:1.Content Based(Eg tags,genre,starring,director) 2.Collaborative Filtering(Item to item based/User to User based)

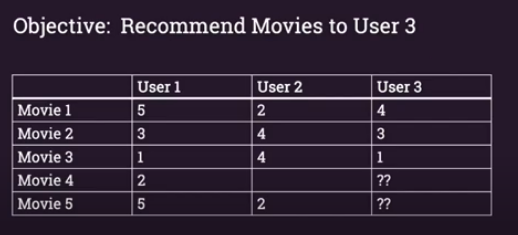
**Collaborative Filtering**

**[4]**

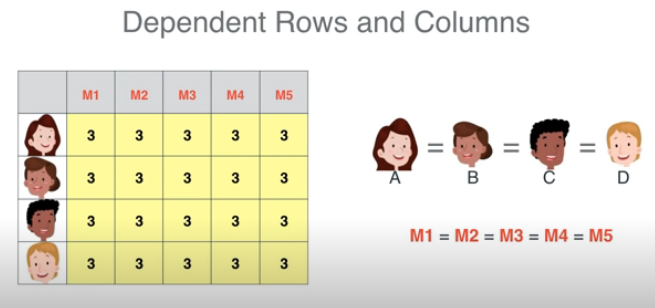
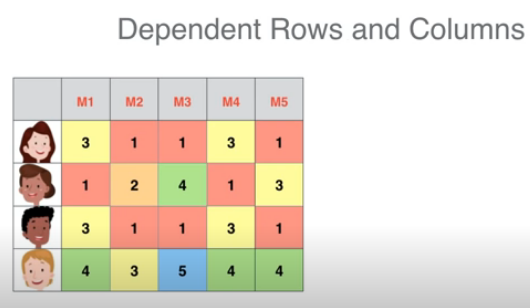
1.Item to Item-finding similar movies based on user rating



2.User to User- Find similar users based on how they rated movies



How does it work?

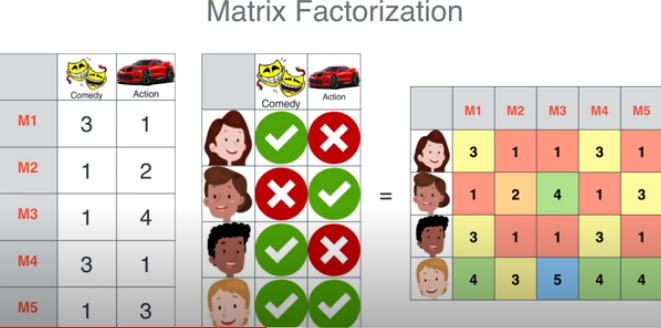
Above, we have two datasets ,we have 4 users and 5 movies ,and each cell represents the rating that the user gave the movie. It is quite obvious that the first dataset is not realistic as every user has given the same rating for every movie.

Now consider the second table, if we observe closely the first and 3rd row are the exact same, the first and third column are also same ,the row 4 can be obtained b also same ,the row 4 can be obtained by adding row 1 to row 2.Therefore it can be said that certain correlations exist between the rows and the columns.

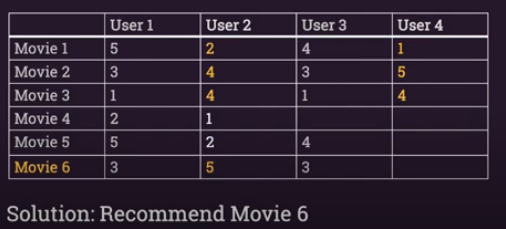
Now comes the Linear Algebra or to be more precise matrix factorisation!

What is factorisation ? example 24 is also 6\*4

Therefore the idea here is to represent our dataframe in terms of two smaller matrices ,such that we can extract only those features of the bigger dataset that are most relevant to our problem statement.

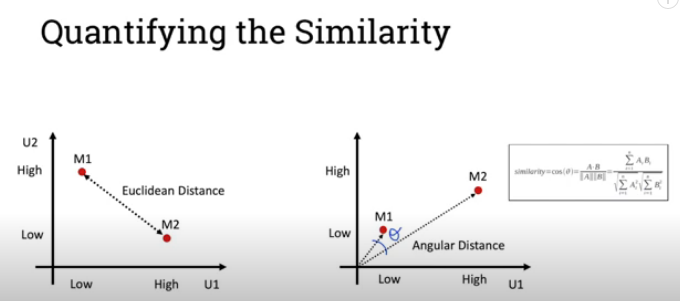
 

Now consider this table:



How would user 4 rate movie 6?

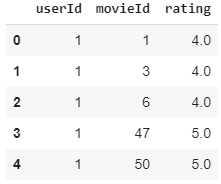
If we analyse the table user 2 and user 4 have rated all movies similarly so based on this correlation we can predict that user 4 would rate movie 6 as 5.



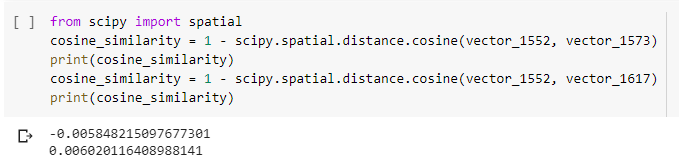
Once we have established our feature ,we can convert it to a vector and based on cosine similarity we can predict how one movie is similar to the other based on user rating.

Here we perform factorisation using singular value decomposition on a movie lens dataset.

**Results and discussion:**

 snippet of our dataset

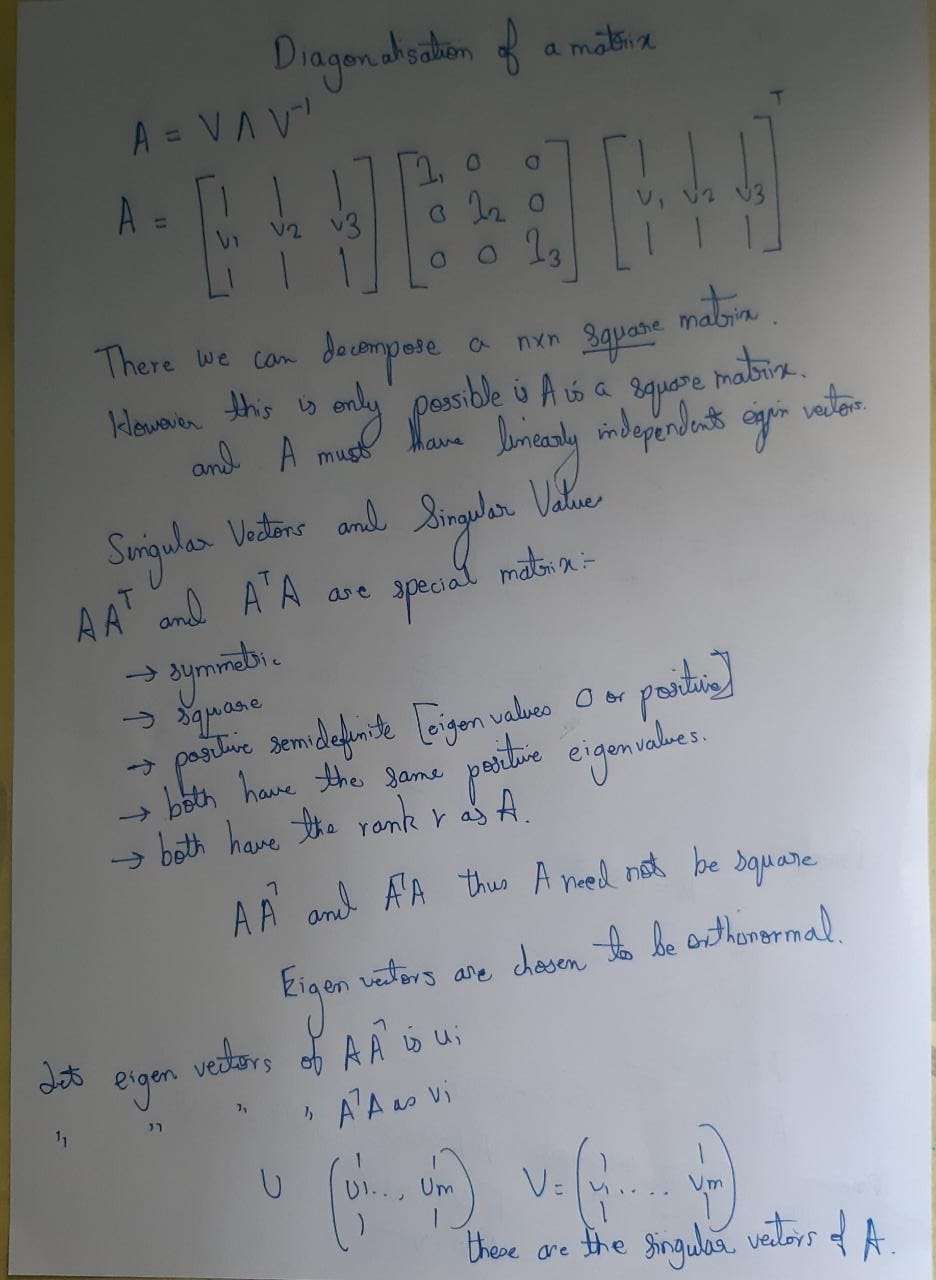
We perform collaborative filtering and extract the feature vector

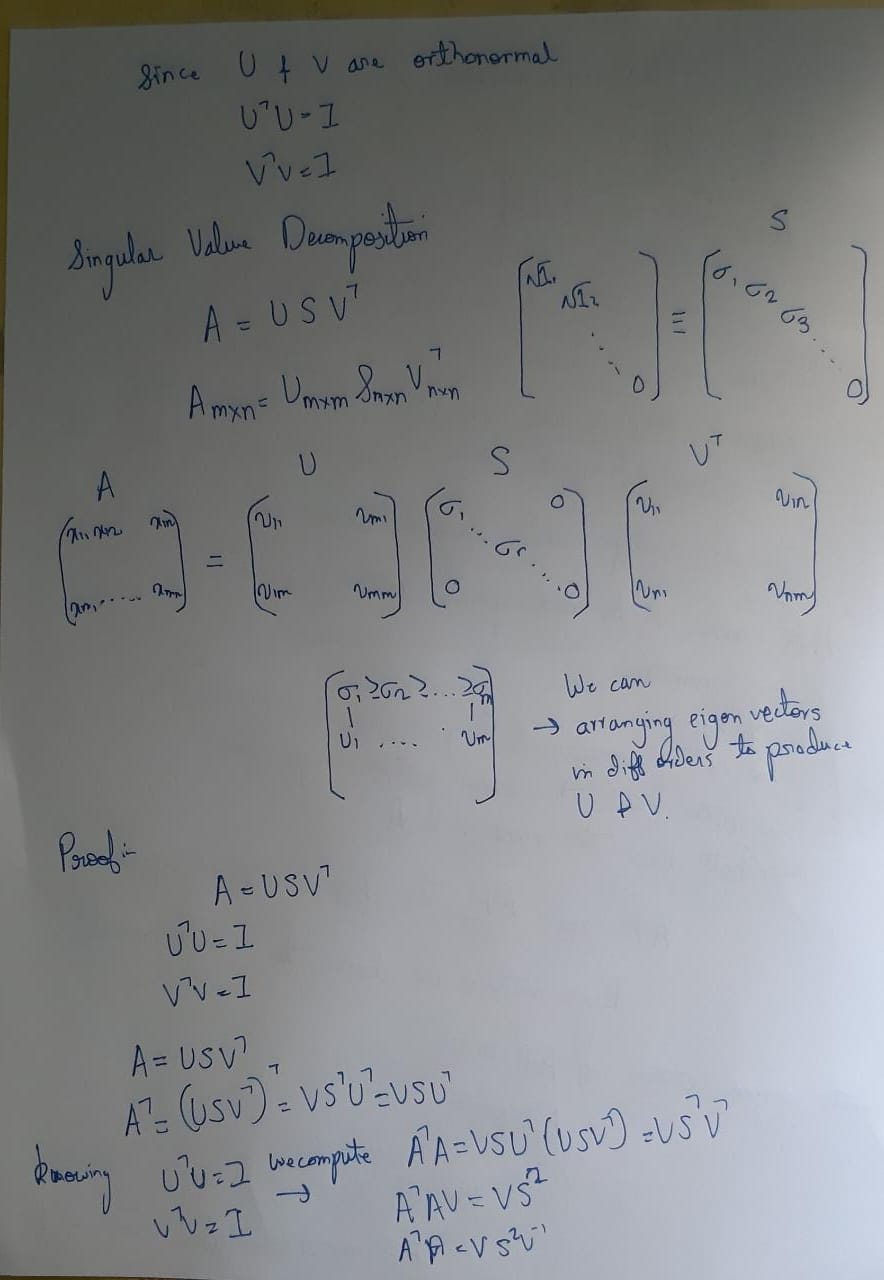


If you observe movie with id 1552 and 1573 have negative cosine similarity hence not at all similar where as movie with id 1552 and 1617 have cosine similarity closer to 1 hence they can be said to be more similar .

Now for the mathematics behind singular value decomposition[6.3]

Derivations and formulae[]





**Summary/Conclusion:**

We thus see how support vector decomposition can be used for predictions based on user rating and this algorithm is used globally by major service providers like Amazon,Netlfix,Spotify that attend to billions of people using Singular Value Decomposition.

**SECTION C: Image Compression using Singular Value Decomposition.**

**Introduction:**

From our derivation in Section B it is understood that SVD can decompose or factorise a matrix into smaller matrices and therefore can be used to reduce the dimensions of a matrix.[7]

**Report on present investigation**:

The basic concept is to represent an image with size m by n as a two-dimensional m by n matrix. SVD is then applied to this matrix to obtain the U, S, and V matrices. S is a diagonal m by n matrix whose number of non-zero elements on the diagonal determine the rank of the original matrix. The fundamental concept of the SVD-based image compression scheme is to use a smaller number of rank to approximate the original matrix.

Quality of the compression is measured in term of Peak Signal to Noise Ratio, which is defined as:[5]  
  
 **PSNR = 10 log10((max. range)2 / Root Mean Square Error )**

**Matlab Code:**

clc;

clear all;

A=imread('H:\PESU\4th sem\Linear Algebra\donald\_trump.jpg');

A=rgb2gray(A);

%imshow(A);

I=im2double(A);

[U,S,V]=svd(I);

S2=S;

S2(20:end, :)=0;

S2(:,20:end)=0;

D=U\*S2\*V';

imshow(D);

Z=imabsdiff(D,I);

imshow(Z);

peaksnr=psnr(D,I);

disp(peaksnr);

**Output/Result**



Psnr= 30.2582

Here k value is 20 which means only 20 singular values were used to reconstruct the image.

**Summary/Conclusion**

Dimensionality reduction using SVD can be thus used to compress a digital image and thus occupy less space.

**Bibliography:**

1.Linear Algebra and its Applications Gilbert Strang.4th edition

Chapter 3 Sections:3.2 3.3

Chapter 5 Sections:5.2

Chapter 6 Sections:6.3

2.<https://medium.com/@andrew.chamberlain/the-linear-algebra-view-of-least-squares-regression-f67044b7f39b>

3.<https://medium.com/@jonathan_hui/machine-learning-singular-value-decomposition-svd-principal-component-analysis-pca-1d45e885e491>

4.<https://realpython.com/build-recommendation-engine-collaborative-filtering/>

5.<https://in.mathworks.com/help/images/ref/psnr.html>

6.<https://in.mathworks.com/help/images/ref/imabsdiff.html>

7.<http://fourier.eng.hmc.edu/e161/lectures/svdcompression.html>