THE APPLICATION OF LINEAR ALGEBRA IN MACHINE LEARNING

Khadeejah James Audu[*1], Oluwatobi Oluwaseun Oluwole[2], Yusuph Amuda Yahaya[3] and Samuel David Egwu[4]

[1,2]Department of Mathematics, Federal University of Technology, Minna, Niger State, Nigeria.
[3]Department of Mathematics, Pen Resource University, Gombe, Gombe State, Nigeria
[4]Department of Computer Science, Federal University of Technology, Minna, Niger State, Nigeria.

emails: [1]k.james@futminna.edu.ng; [2]oluwole.m1701595@st.futminna.edu.ng.com;
emails: [3]yusuph.yahaya@pru.edu.ng; [4]egwusamuel2015@gmail.com

Abstract

In the realm of machine learning, incorporating linear algebraic methods has become indispensable, serving as a foundational element in developing and refining various algorithms. This study explores the significant impact of linear algebra on machine learning applications, highlighting its fundamental principles and practical implications. It delves into key concepts such as vector spaces, matrices, eigenvalues, and eigenvectors, which form the mathematical basis of well-established machine learning models. The research provides a comprehensive overview of how linear algebra contributes to tasks such as classification, regression analysis, and dimensionality reduction. It also investigates how linear algebra simplifies data representation and processing, enabling effective handling of large datasets and identification of meaningful patterns. Additionally, the study explores specific machine learning applications like Word/Vector Embedding, Image Compression, and Movie Recommendation systems, demonstrating the critical role of linear algebra. Through case studies and practical examples, the study illustrates how a deep understanding of linear algebra empowers machine learning practitioners to develop robust and scalable solutions. Beyond theoretical frameworks, this research has practical implications for practitioners, researchers, and educators seeking a deeper understanding of the relationship between machine learning and linear algebra. By elucidating these connections, the study contributes to ongoing efforts to improve the efficacy and efficiency of machine learning applications.

KEYWORDS: Linear Algebra, Machine Learning Applications, Mathematical Underpinnings, Data Representation, Model performance.

1 INTRODUCTION

Since its inception in the early 1700s for the purpose of solving linear equations, linear algebra has undergone significant evolution. It has evolved into a critical branch of mathematics essential for data analysis and problem-solving across various domains. Particularly within the realms of data science and machine learning (ML), its importance has been underscored as a foundational requirement (Al-Thaedan *et al.*, 2024). Linear algebra's significance has grown notably in recent times, especially due to its indispensable role in understanding and optimizing machine learning techniques. In machine learning, a subset of artificial intelligence, algorithms derive insights from data to make informed predictions or decisions. Linear algebra provides the mathematical framework necessary for data representation, manipulation, and modeling, thereby facilitating complex problem-solving in machine learning. Data is commonly represented using matrices and vectors in the context of machine learning, enabling manipulation through a range of linear algebraic operations. These operations, which constitute the fundamental components of machine learning algorithms, include computations of eigenvalues, matrix multiplications, and inner products. Moreover, linear algebra simplifies tasks such as dimensionality reduction, model optimization, and data transformation, thereby aiding in the advancement of machine learning practices.

The integration of linear algebra in machine learning presents a significant gap in current research due to the lack of comprehensive studies that elucidate its practical implications and applications. This research aims to address this gap by providing a thorough exploration of how linear algebra techniques can be effectively applied in various machine learning tasks. The novelty lies in the comprehensive examination of linear algebra's role in tasks such as classification, regression analysis, and dimensionality reduction, shedding light on its fundamental principles and practical significance. This study is motivated by the increasing demand for robust and scalable machine learning solutions, where a deeper understanding of linear algebra can significantly enhance model performance and efficiency. The implications of this research extend to both academia and industry, offering valuable insights for researchers, practitioners, and educators seeking to improve the efficacy and efficiency of machine learning applications through the application of linear algebraic methodologies.

1.1 LITERATURE REVIEW

Audu (2022) Examines enhancements and practical applications of Singular Value Decomposition (SVD) in image compression within the domain of machine learning. It elucidates the SVD concept and demonstrates its implementation for compressing images in the Python environment. Recent research (R. SINIR and B SINIR, 2019; Wicaksono and Setiawan, 2022) suggests various approaches to designing movie recommendation systems utilizing the SVD algorithm. These systems leverage dimensionality reduction to handle sparse data, enabling personalized film recommendations based on user input parameters. By combining collaborative filtering with SVD and Support Vector Machine (SVM) classification, the study develops a user-centric recommendation system utilizing tweet evaluations, adaptable to both user- and item-based datasets. Another study (Ramni *et al.*, 2020) proposes a method for offering diverse film suggestions considering factors like popularity and genre. Utilizing deep learning techniques and cosine similarity, this method implements a Content-Based Recommender System, while also addressing challenges encountered by such systems. Additionally, (Alshamman, 2023) employs

Python's cosine similarity and word frequency functions to create a text comparison program, focusing on measuring text similarity using cosine similarity metrics.

2 MATERIALS AND METHODOLOGY

2.1 Description of Machine Learning

Machine learning, a subset of artificial intelligence, relies on foundational advances in mathematics and computer science to develop algorithms that enable computers to learn from data and make predictions or decisions autonomously. Supervised learning involves training algorithms on labeled datasets, where each example is paired with a corresponding label or outcome, allowing the algorithm to learn the mapping between inputs and outputs. Semi-supervised learning improves model performance by leveraging both labeled and unlabeled data, while unsupervised learning uncovers hidden patterns or structures within unlabeled data using clustering and dimensionality reduction techniques.

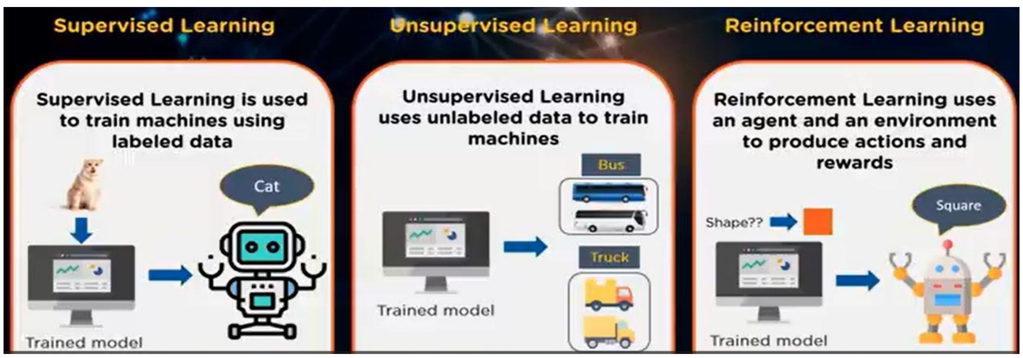

Figure 1. Types of Machine Learning

The machine learning pipeline consists of key stages such as data collection, preprocessing, feature engineering, model selection, training, evaluation, and deployment, each crucial for ensuring the quality and effectiveness of the model. Continuous exploration of innovative approaches like data augmentation, transfer learning, and robust evaluation metrics enhances various aspects of the process. Machine learning applications span diverse domains, including healthcare for medical image analysis, finance for fraud detection and stock market prediction, and natural language processing for language translation, sentiment analysis, and chatbot development. Its versatility makes machine learning a cornerstone of modern technological advancements, driving innovation across industries and domains.

Figure 2. Machine Learning Approach

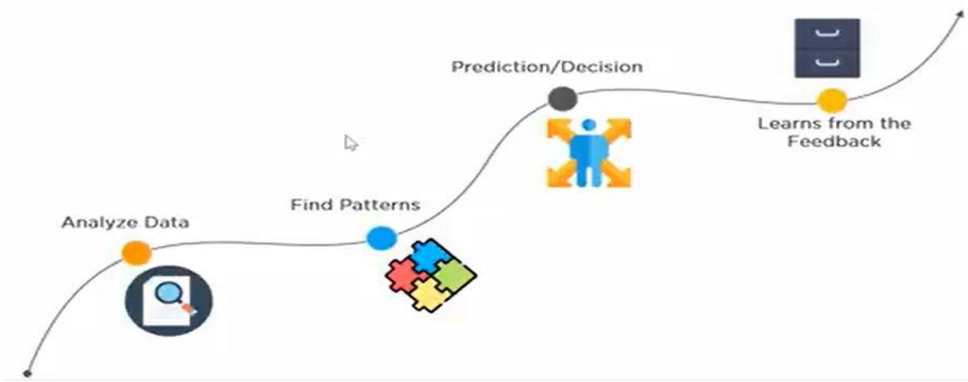

Figure 3. Machine Learning Process

2.2 Linear Algebra Techniques

2.2.1 Evaluation of Eigenvalaue and Eigenvector

Eigenvalues and eigenvectors are fundamentals in linear algebra and play important roles in many other domains. They are related to square matrices and offer important insights into how vector spaces behave and change (Savant, 2020). The matrix below serves as an illustration of this.

$$B = \begin{bmatrix} 4 & 6 & 10 \\ 3 & 10 & 13 \\ -2 & -6 & -8 \end{bmatrix}$$

The eigenvalues of B is calculated using the characteristic equation formula

$$|B - \lambda I| = 0 \tag{1}$$

Which results to $\lambda_1 = 0,\ \lambda_2 = 2,\ and\ \lambda_3 = 4$. Consequently, matrix A's eigenvalues are 0, 2, and 4. Assuming a 2x2 matrix, it is possible to obtain the eigen vectors as follows

$$C = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}$$

The characteristics equation is evaluated by the expression $\det(C - \lambda I) = 0$

$$\det\left(\begin{bmatrix} 3-\lambda & 1 \\ 1 & 2-\lambda \end{bmatrix}\right) = 0$$

The two eigenvalues are $\lambda_1 = 4\, and\ \lambda_2 = 1$. The corresponding eigen vectors are evaluated as;

$$\text{for } \lambda_1 = 4;\ |C - 4I| = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \Rightarrow v_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Solving for eigenvectors gives

For $\lambda_2 = 1$; $|C - I| = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \Rightarrow v_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$. Therefore, the eigenvectors corresponding to the eigenvalues $\lambda_1 = 4 \, and \; \lambda_2 = 1$ are $v_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \, and \; v_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$ respectively.

2.2.2 Matrix's Diagonalization

A vector space V is a collection of vectors and two operations which are vector addition and scalar multiplication which hold for a, b, c ∈V and x, y ∈ $\Re$ which obeys certain prescribed laws. Matrices is a representation of vectors in rows and columns. Diagonalization of matrices which is an essential in linear algebra that we will also be looking at enables us to simplify and analyze matrices well. (Aggarwal, 2020; Chollet, 2018). Finding a diagonal matrix and an equivalent invertible matrix that, when multiplied together, produce the original matrix is the first step in diagonalizing a matrix. The steps to check if a matrix is diagonal are as follows.

Step1: Evaluate the eigenvalues of any matrix, such as B.
Step 2: Verify that no effective is flawed, as this will prevent the matrix from diagonalizing. If not, you can move on to the following phase.
Step 3: Locate as many linearly independent eigenvectors as you can for each eigenvalue; the number of these eigenvectors is equal to the eigenvalue's geometric multiplicity.
Step 4: Adjoin all the eigenvectors so as to form a full-rank matrix P.
Step 5: Build a diagonal matrix D whose diagonal element are the eigenvalues of B.
Step 6: The diagonalization is done $D = P^{-1}BP$

2.2.3 Singular Value Decomposition (SVD)

The matrix factorization, which divides a matrix into two or more matrices, is an effective data analysis and collaborative filtering tool. It captures underlying relationships, allows prediction of missing values, and is used in the context of recommendation systems to reveal latent patterns in user-item interaction data by factorizing the original matrix. The reverse of matrix multiplication is matrix factorization. SVD is the name for matrix factorization. The principle of eigenvalue decomposition of complex matrices is essentially generalized by the SVD. (Audu, 2022; Karthigai and Selvam, 2021). In linear algebra, decomposition of any matrix may only be applied to a diagonalizable square matrix, which is a type of square matrix. Any given matrix's SVD can be factored into smaller matrices, as shown below, using the expression.

$$P_{m \times n} = R_{m \times m} \sum_{m \times n} S^T{}_{n \times n} \tag{2}$$

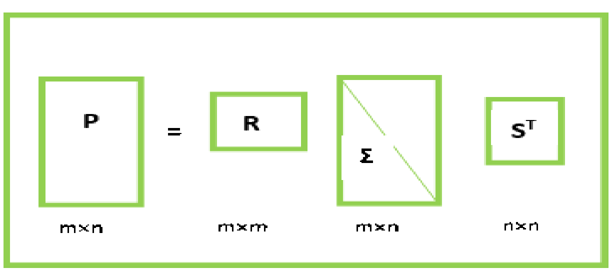

Figure 4. Schematic Representation of SVD

where P is a matrix, R is a m×m orthogonal matrix, $\sum$ is the diagonal matrix and S^T is a n×n orthogonal matrix with m as the row, n the column associated to the matrices. For the proper application of the SVD technique to reduce high dimensionality, one can demonstrate how it's been computed using a 3×2 matrix P

$$P = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ -1 & 1 \end{bmatrix}$$

by following some few steps to factor matrix P into $R\sum S^T$. The sequence of steps are :

Step 1: Construct matrix $P^T P$ of a real m×n matrix P.

Step 2: Compute the eigenvalues of $P^T P$.

Step 3: Compute the matrix S^T.

Step 4**:** Compute the $\sum$ matrix.

Step 5: Compute the matrix $P \bullet P^T$.

Step 6: Compute the eigenvalues for $P \bullet P^T$.

Step 7: Compute the matrix R.

Step 8: Write the matrix P into the SVD.

The resultant decomposition after applying the above SVD steps gives

$$P = \begin{bmatrix} 1/\sqrt{3} & 1/\sqrt{2} & 1/\sqrt{6} \\ 1/\sqrt{3} & 0 & -2/\sqrt{6} \\ 1/\sqrt{3} & -1/\sqrt{2} & 1/\sqrt{6} \end{bmatrix} \begin{bmatrix} \sqrt{3} & 0 \\ 0 & \sqrt{2} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = R\sum S^T$$

SVD is a technique in linear algebra that can be used for compressing image. Images can be viewed as arrays of numbers in large matrix and could be in shades of various color. Having an image of 300×200 pixels, this is computed to 6000-pixel numbers for the image storage. When it is splitted to red, blue and green layers the number entries become 18000. The image can be reduced by a 5-term SVD, enabling the storage of 5 sigma (σ) and five thousand each of the $R\, and\, S^T$. This reduces the storage cost by and this implies that a matrix can be represented for few numbers using SVD. (Kalaivazhivijayaragavan *et al.*, 2022; Zaki *et al.*, 2024) By the low rank approximation, we approximate our, $P(K) = \sum \sigma_r R_r S_r^T$ matrices instead of $P = R\sum S^T$, where, σ_r is a scalar, R_r is a row vector the matrix R and $S^T{}_r$ is a column vector the matrix S^T. The rate at which the SVD compresses the image can be measured through the calculations of the compression ratio and this is gotten using the formula below

$$G_r = \frac{n \times m}{q(n+m+1)} \tag{3}$$

where is G_r the compression ratio, n is the row, m is the column and q is the rank. Also, the mean square error (MSE) is used to compare the quality of the uncompressed (P) to the compressed (P_r). This can be computed by the relation

$$MSE = \frac{1}{nm}\sum_{a=1}^{n}\sum_{b=1}^{m}\left[h_a(a,b) - h_{a_r}(a,b)\right]^2 \tag{4}$$

2.2.4 Cosine Similarity

Cosine similarity is a mathematical measure used to access the similarity between two vectors in an n-dimensional space. For vector A and B, it is calculated as the cosine of the angle θ between them, expressed as the dot product of A and B divided by the product of their magnitude (Ramni *et al.*, 2020). Figure 5 shows the illustration of cosine similarity

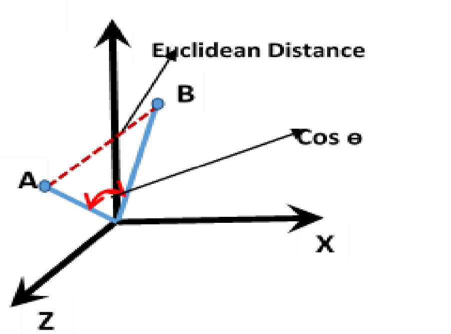

Figure 5. Diagram of Cosine Similarity

The application of cosine similarity to our study will be incomplete without showing how to obtain a cosine similarity. Having two vectors U_a and U_b, the following procedure shows the use of cosine similarity to find their similarity.

$$U_a = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix} \quad U_b = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}$$

Cosine similarity procedure is analysed in the following manner;

$$(U_a \bullet U_b) = \frac{U_a \bullet U_b}{\|U_a\| \bullet \|U_b\|} \tag{5}$$

$U_a \bullet U_b$ is the dot product of the vectors while $\|U_a\| \, and \, \|U_b\|$ are the magnitude of the vectors

Step 1: Calculate the dot product of $U_a \bullet U_b$

$U_a \bullet U_b = (2\times1)+(3\times2)+(1\times2)=10$

Step 2: Calculate the magnitude of S_a and S_b

$$\|U_a\| = \sqrt{(2)^2 + (3)^2 + (1)^2} = \sqrt{4+9+1} = \sqrt{14}$$

$$\|U_b\| = \sqrt{(1)^2 + (2)^2 + (2)^2} = \sqrt{1+4+1} = \sqrt{9}$$

Step 3: Substitute these values into the cosine similarity formula

Cosine similarity formula $(U_a \bullet U_b) = \dfrac{U_a \bullet U_b}{\|U_a\| \bullet \|U_b\|} = \dfrac{10}{\sqrt{14} \times \sqrt{9}} = \dfrac{10}{\sqrt{126}} = 0.07937$

The result here shows dissimilarity because it’s close to 0 but if the result for a cosine similarity is close to 1 or its 1, then it shows similarity or implies that there is similarity.

2.2.5 REPRESENTATION OF LINEAR ALGEBRA

In the realm of machine learning, data is converted into vectors to enable computational processing. Within this domain, terminology such as "feature vector" and "feature space" is frequently utilized (Chollet, 2018; Anichur *et al.*, 2024). A feature vector is a representation of an object's attributes, with each component of the vector corresponding to a specific feature. On the other hand, a feature space encompasses these feature vectors. An exemplar of this concept is provided below.

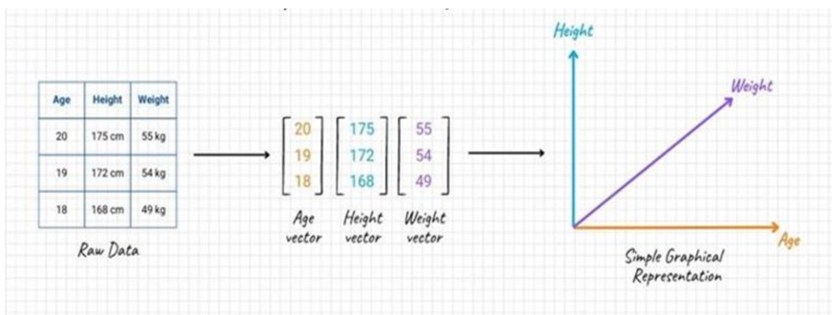

Figure 6. Diagram Utilizing Vectors and Matrices for Features

A matrix format can also depict images. In a black and white image, pixels are represented by 0s and 1s, as demonstrated in the example image below.

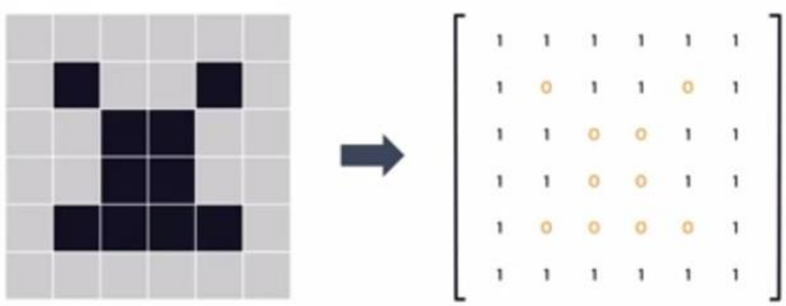

Figure 7. Image Diagram Using Matrices

Regarding terms and documents, a method involves assigning each term a vector where each entry corresponds to the frequency of the word's occurrence in a particular document. These vectors collectively form a matrix useful for analysis. For example, to determine the frequency of the term "man," its occurrences can be counted across five different publications.

$$
man \begin{bmatrix} 0 \\ 7 \\ 0 \\ 51 \\ 15 \end{bmatrix} \begin{array}{l} Article\,1 \\ Article\,2 \\ Article\,3 \\ Article\,4 \\ Article\,5 \end{array}
$$

For yes or no ratings, each movie can be linked to a feature vector, with dimensions representing different attributes. These vectors enable mathematical operations such as similarity measurement and personalized recommendations based on user preferences and movie features. This user feedback system effectively communicates preferences and opinions regarding movies or products. Representing ratings as vectors enhances their usefulness in data analysis and recommendation systems. The relationship between the viewer, movie title, and its vector representation is illustrated.

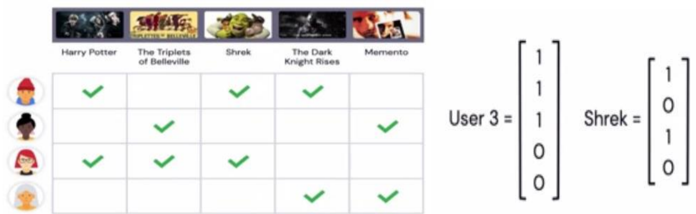

Figure 8. Vector Diagram of Movie to User Synergy

3 APPLYING LINEAR ALGEBRA IN MACHINE LEARNING

Experiment 1: We investigated the compression of an image in this study, focusing on the image of a Hen with dimensions 3456 × 5184. Using a linear algebra technique known as SVD, the objective is to compress the Hen image without significant loss of information. The procedure involves decomposing the image matrix into its singular values and vectors, detailed in Steps 1 through 8 of Section 2. The results, depicted in Figures 10 through 15, illustrate the application of SVD to reduce the original Hen image using ranks ranging from 25 to 150.

Figure 9. Actual Image

Figure 10. q = 25

Figure 11. q = 50

Figure 12. q = 75

  

Figure 13. q = 100 **Figure 14**. q = 125 **Figure 15**. q = 150

Findings: From the provided images, it can be noted that at q = 25, the image appears significantly blurred yet retains important details. Moving to q = 50, although still somewhat blurry, there's noticeable improvement compared to q = 25. As the values of q increase from 75 to 150, the images become progressively clearer, indicating that clarity improves with higher singular values.

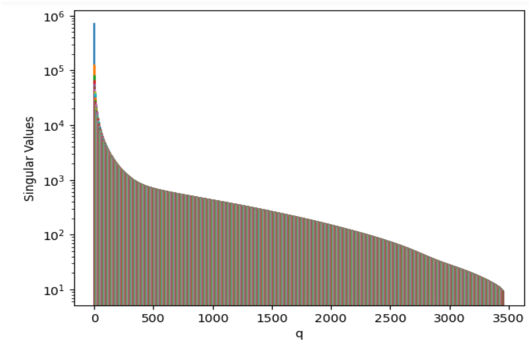

Figure 16. Singular Values Effects concerning Hen

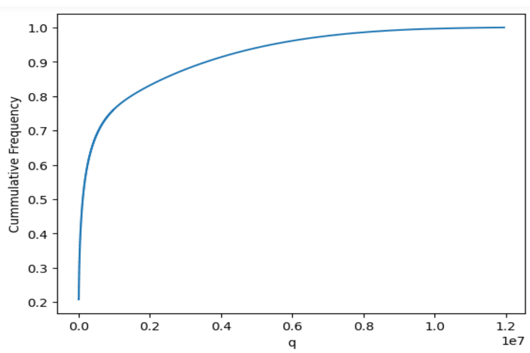

Figure 17. Summation concerning Hen Singular Compression Values

Figure 13 illustrates the relationship between the number of singular values used in image compression and the Mean Squared Error (MSE) of the resultant image. As the count of singular values increases, the MSE decreases, indicating an enhancement in image fidelity. However, this improvement eventually reaches a point of diminishing returns, leading to a less significant enhancement in image quality while simultaneously increasing file size. This trend suggests an optimal balance between the number of singular values utilized and the resulting image quality. In Figure 14, the cumulative sum of singular values for the Hen image is displayed, presenting data points specific to observations of the Hen. The curve demonstrates a gradual tapering from an initially steep slope, suggesting that the initial singular values carry more significance compared to subsequent ones.

Table 1: **Outcome of Hen SVD Image Compression**

Singular Values (q)	Storage Space (kb)	Compression Ratio	Mean Square Error
25	125	1.20577225687	338.409177175
50	146	2.41154451374	157.335607123
75	150	3.61731677061	95.0764210577
100	155	4.82308902749	65.4949112996
125	159	6.02886128436	49.1690764436
150	162	7.23463354123	39.20512936439
Original	2.24 MB	1	

The presented table offers a comparison of image compression results for the Hen using various compression ratios. Each compression ratio includes details such as the original file size, compressed file size, compression ratio, and Mean Square Error (MSE). As the compression ratio increases, the file size grows, while the mean square error decreases, leading to enhanced image quality. The table aims to strike a balance between reducing file size and maintaining image quality, aiding in the selection of an optimal compression ratio. Additionally, it illustrates how the original images can be compared to the compressed ones and emphasizes the use of lower Mean Square Errors (MSE) for achieving superior image quality.

Experiment 2: This experiment aims to recommend movies for User 65, leveraging a dataset with 10,000 users and 3,883 movies from the Alyssaq Dataset. Using Singular Value Decomposition (SVD), the dataset's dimensions are reduced to 3,952 by 3,952, and cosine similarity within an Item-Based Collaborative Filtering Recommending System is utilized to select a movie for User 65. Initial data processing involves creating a new matrix based on Tables 2 and 3, with movie titles serving as vectors for similarity checks. Through Steps 1 to 5 under Cosine Similarity, unwatched movies most similar to those already rated by User 65 are recommended. The results, presented in Tables 2 to 4 and Figure 15, demonstrate the achievement of the goal.

Table 2: Display of Ratings and Movie ID

Index	Movie_ID	User_ID	Rating	Time
0	1	1193	5	978300760
1	1	661	3	978302109
2	1	914	3	978301968
3	1	3408	4	978300275
4	1	2355	5	978824291
5	1	1197	3	978302268
6	1	1287	5	978302039
7	1	2804	5	978300719
8	1	594	4	978302268
9	1	919	4	978301368
⋮	⋮	⋮	⋮	⋮

9999	3883	3246	4	987302019

Table 2 presents the initial ten films from the dataset, showcasing their respective indices. Each user in the dataset is identified by a unique User ID, and their reviews for each movie are represented by a Movie ID, accompanied by ratings ranging from 1 to 5 and timestamps derived from the movie dataset (Asoke et al., 2018).

Table 3. Tabular presentation featuring Movie ID, Title, and Genre

Index	Movie_ID	Title	Genre
0	1	Toy Story (1995)	Animation/Children's/Comedy
1	2	Jumanji (1995)	Adventures/Children's/Fantasy
2	3	Grumpier Old Men (1995)	Comedy/Romance
3	4	Waiting to Exhale (1995)	Comedy/Drama
4	5	Father of the Bride Part II (1995)	Comedy
5	6	Heat (1995)	Action/Crime/Thriller
6	7	Sabarina (1995)	Comedy/Romance
7	8	Tom and Huck (1995)	Adventure/Children's
8	9	Sudden Death (1995)	Action
9	10	Golden Eye (1995)	Action/Adventure/Thriller
⋮	⋮	⋮	⋮
3882	3883	Flying Saucer (1950)	Sci-Fi

Table 3 displays the initial ten films from the dataset along with their respective genres, featuring titles such as "Jumanji," "Toy Story," and "Heat." The genres encompass a wide range, from humor and animation to action and thrillers. Notably, many films span multiple genres; for instance, "Toy Story" is categorized under Comedy/Children's/Animation. Given our use of the item-based collaborative filtering approach, both the title and genre are included to facilitate comprehension and analysis of the results. Additionally, the table incorporates Movie ID and Index identifiers for reference.

Table 4. Suggested Movies Pivot Table

User	Toy Story	Father's Mind	Grumpier	The Sailor	Heat	⋯
User 1	5	5	4	3	5	⋯
User 2	3	4	4	3	4	⋯
User 3	4	3	4	5	3	⋯
User 4	5	5	4	4	5	⋯
User 5	3	4	3	5	5	⋯
User 6	4	4	5	4	3	⋯
User 7	5	3	3	3	5	⋯
User 8	3	3	5	3	4	⋯
User 9	4	5	2	5	3	⋯
User 10	5	3	5	5	4	⋯

⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 4 represents a matrix comprising all movies viewed by users and their respective ratings, demonstrating the connections between movies based on the viewership patterns across all users. Each row corresponds to a user, and each column corresponds to a movie title, with rating values present in each cell. By synthesizing information from Tables 2 and 3, a new matrix of dimensions 3,883 by 10,000 was generated. Application of SVD compresses this matrix to a size of 3,952 by 3,952, resulting in Table 4. Since the recommendation system relies on item-based collaborative filtering, the similarity between movies is assessed based on the movie title column, where each column serves as a vector. For instance, the "Toy Story" column is treated as a vector and compared with other films using the title column in Table 4, employing cosine similarity to gauge he likeness between movies.

```
Recommendations for Toy Story (1995):

1.  Toy Story (1995)
2.  Spellbound (1945)
3.  Gay Divorcee, The (1934)
4.  Hurricane Streets (1998)
5.  Mystery Science Theater 3000: The Movie (1996)
6.  Friday (1995)
7.  Canadian Bacon (1994)
8.  Endless Summer, The (1966)

Index number of the recommended movies:  [   0  930  906 1658  670   68  156 3652]
```

Figure 18. Result for Movie Recommendation

The above image showcases the outcome of recommending the top eight films similar to Toy Story (movie ID 1) for User ID 65, who has viewed the movie and rated it five stars.

Experiment 3: This experiment focuses on word embedding, employing cosine similarity to assess similarity among four sentences: 1, 2, 3, and 4. Each sentence is represented as a vector in this analysis. The sentences are as follows:

1. "Mathematics is a core discipline for academics"
2. "Mathematics is a course in my school"
3. "I love my course"
4. "Dr K. J is a lecturer in my school"

The objective is to demonstrate the practical application of linear algebra tools, specifically cosine similarity, in determining the similarity between sentences within the Python environment. The procedure involves encoding the phrases as numerical vectors, calculating the cosine similarity between these vectors, and interpreting the resulting score as a measure of similarity. The process follows Steps 1 through 5 outlined in Section 2.2 under Cosine Similarity. The results, presented in Table 5 and Figure 16, offer a graphical representation of the cosine similarity between the sentences.

Table 5: Cosine Similarity Findings

Sentence Pairs	Cosine Similarity
Sentence 1 and Sentence 2	0.6667
Sentence 1 and Sentence 3	0.2583
Sentence 1 and Sentence 4	0.6695
Sentence 2 and Sentence 3	0.6395
Sentence 2 and Sentence 4	0.6898
Sentence 3 and Sentence 4	0.4562

As per the table, Sentence 1 and Sentence 4 exhibit the highest similarity (0.6695), followed by Sentence 2 (0.6667). This indicates that while Sentences 1 and 2 share some similarity, Sentences 1 and 4 convey comparable meanings. Sentence 3 appears to be the least similar to the others, implying a significant difference in meaning compared to the other sentences. Sentences 2 and 4 emerge as the most similar pair, while Sentence 3 stands out as the most distinct. Overall, the table reveals intriguing patterns of similarity among the sentences.

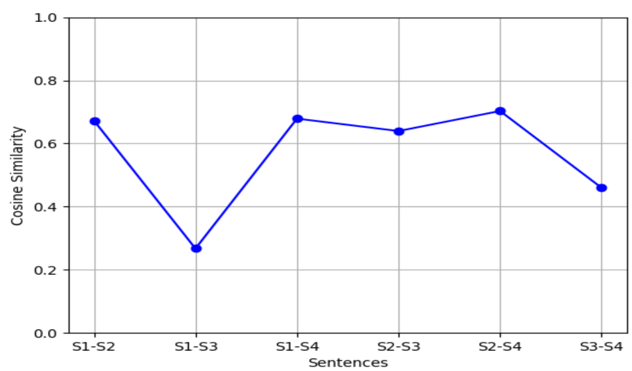

Figure 16. Visual depiction demonstrating cosine similarities.

The graph depicts the correlation between sentence pairs along the horizontal axis and cosine similarity along the vertical axis. Notably, sentences 1 through 3 are positioned closest to 0, indicating the least resemblance, whereas sentences 2 through 4 are closer to 1, indicating higher similarity. Among them, sentences 2 and 4 exhibit the highest cosine similarity value of 0.6898, suggesting they are the most similar sentences in the set.

4 CONCLUSION

This study has delved into the profound relationship between machine learning and linear algebra, emphasizing the pivotal role that mathematical principles play in augmenting the capabilities of learning algorithms. As depicted in Tables 1, our investigation has yielded superior picture compression in terms of size and reduced errors. Employing an SVD approach within a Python environment, we successfully compressed photos, achieving a higher compression rate while preserving the quality of the original image. Furthermore, we reduced the dimensionality of a specific movie dataset using SVD and utilized cosine similarity to recommend movies based on their similarity, as illustrated in Figure 15. Lastly, cosine similarity was employed to assess similarities between sentences, as evidenced in Table 5 and Figure 18.

Acknowledgments

We extend our heartfelt thanks to all contributors, funders, and supporters who made this research possible.

REFERENCES

Aggarwal, C. C. (2020). *Linear Algebra and Optimization for Machine Learning*. Springer Nature Switzerland AG. 1-478.

Alshamman, A. F. (2023) Implementation of Text Similarity Using Word Frequency and Cosine Similarity in Python. *International Journal of Computer Applications, 185*(36), 54-59.

Al-Thaedan, A. Z. Shakir & Mjhool A. Y. (2024) A Machine Learning Framework for Predicting Downlink Throughout in 4G-LTE/5G Cellular Networks. 16(6), 651–657.

Anichur, R., Tanoy D., Kundu D., Khan S. I., Airin A. A. A., Sadia S., Sayduzzaman M. & Shahab S. B. (2024). Machine learning and deep learning-based approach in smart healthcare: Recent advances, applications, challenges and opportunities. *AIMS Public Health*. 11(1), 58-109.

Asoke, N., Adityam, G., & Ariyon, M. (2018). Building a Movie Recommendation System using SVD algorithm.International Journal of Computer Sciences and Engineering, 6 (11), 727-729.

Audu, K. J. (2022). Application of Singular Value Decomposition technique for compressing Images. *Gadau Journal of Pure and Allied Sciences*, *6*(2), 82-94.

Habibi. M. A, Fakhfouri A., Mirjani M. S. (2024) Prediction of Cerebral Aneurysm Rupture Risk by Machine Learning Algorithms: A Systematic Review and Meta-Analysis of 18,670 Participants. *Neurosurgical Review*. 47(1), *34*.

Halim, S. (2020). Application of Linear Algebra in Machine Learning. *International Research Journal of Engineering and Technology, 7*(2), 3188-3197.

Kalaivazhivijayaragavan, Abinisha, K., Dhara, T., & Santhiya, E. (2022). Machine Learning Approach for Linear Algebra. *International Research Journal of Modernization in Engineering Technology and Science, 4*(5), 3842-3845.

Karthigai, S. S., & Selvam S. (2021) Image Compression Using Linear Algebra with SVD Algorithms. *Asian Journal of Engineering and Applied Technology, 10*(1), 22-28.

Kenneth, B. W. B, El-Shafie A., Othman F., Khan M. H., Ahmed H. B., & Ahmed A. N. (2024). Groundwater level forecasting with machine learning models: A review. *Water Research*. 252.

Kiani, S. M, Belda S., & Mishra S. (2024) Short-term prediction of celestial pole offsets with interpretable machine learning. *Earth Planets Space 76*. 18, 1-15.

Koosha, S., & Mahyar A. (2023) Machine Learning and Deep Learning: A Review of Methods and Applications. *World Information Technology and Engineering Journal*. 10(7), 3897-3904.

Pouya, I, Heris M. N., Amir T. S. C., Somayeh A., & Yoo C. (2022) Sustainable energies and machine learning: An organized review of recent applications and challenges. *Energy*. 266.

Ramni, H. S, Sargam M., Tanish T., Turshar N. & Gaurav S. (2020) Movie Recommendation System using Cosine Similarity and KNN. *International Journal of Engineering and Advanced Technology* (IJEAT). 9(5), 556-559.

Savant, A. P., (2020). A Review of Concepts Concerning Linear Algebra and the Applications of Similarity Transformations. *GRD Journal for Engineering , 5*(12), 1-7.

SINIR, R. & SINIR B. (2019) Stability Analysis of Principal Parametric Resonace of Viscoelastic Pipes by Using Multiple Time Scales. *Bayburt Universitesi Fen Bilimeri Dergisi*. 284(35), 879-891.

Seth, N. (2024) A Predictive Study of Machine Learning and Deep Learning Procedures Chronic Disease Dataset. *Journal of Artificial Intelligence, Machine Learning and Neural Network*. 4(42), 34-47.

Wicaksono, R. A, & Setiawan E. B. (2022) Recommender System Based on Tweets with Singular Value Decomposition and Support Vector Model Classification. *Journal of Computer Science and Informatatics.* 3(4), 294-302.

Zaki, A. M, Khodadadi N., Lim W. H., & Towfek S. K. (2024) Predictive Analytics and Machine Learning in Direct Marketing for Anticipating Bank Term Deposit Subscriptions. *American Journal of Business and Operations Research* (AJBOR). 11(1), 79-88.

Zeynep H. Kilimci, S. A., (2018). Deep Learning- and Word Embedding-Based Heterogenous Classifier Ensembles for Text Classification. *Hindawi, 24*(5), 1-8.