A Practical Investigation of Principal Component Analysis (June 2024)

Aditya Bajracharya[1], Projan Shakya[1]
[1]Thapathali Campus, IOE, TU

Abstract

In this lab exercise, Principal Component Analysis (PCA) is utilized as a powerful tool to uncover the latent structure within a given dataset. Following initial steps of data standardization and covariance matrix computation, PCA facilitates the identification of principal components, offering a concise representation of the dataset's variability. By visually examining the relationships between these principal components and the 'Target (Total orders)' data, we gain a deeper understanding of the dataset's underlying patterns and contributions. Furthermore, the calculation of the proportion of variation explained by each principal component provides a quantitative assessment of their significance. This comprehensive analysis sheds light on the intricate dynamics and inherent structure of the dataset, contributing valuable insights into its characteristics and potential applications.

Keywords

Covariance Matrix, Principal Component, Proportion of Variation, Standardization, Variability

I. Introduction

TODAY'S world of computers and artificial intelligence has revolutionized decision-making processes, thanks to advanced algorithms like decision tree. With the growing demand for AI-driven solutions, the need for efficient, accurate and understandable models is more crucial than ever.

A decision tree classifier is a type of supervised learning algorithm that is used for both classification and regression tasks. It works by splitting the dataset into subsets based on the value of input features, creating a tree-like model of decisions. Each internal node of the tree represents a feature or attribute, each branch represents a decision rule, and each leaf node represents an outcome or class label.

The simplicity and transparency of decision trees make them highly valuable in various applications. They are particularly useful when dealing with complex datasets where the relationships between features and target variables are not easily discernible. Decision trees can capture non-linear patterns and interactions between variables, making them robust and flexible.

However, decision trees also come with their own set of challenges. They are prone to overfitting, especially when dealing with noisy data or when the tree grows too deep. This can result in poor generalization to new data. To address these issues, techniques such as pruning, ensemble methods (like Random Forests and Gradient Boosting), and cross-validation are often employed.

In essence, decision tree classifiers offer a powerful tool for making informed decisions based on data. Their ability to break down complex decision-making processes into simple, interpretable rules makes them a cornerstone of many machine learning systems. By understanding and mitigating their limitations, we can harness

* Corresponding authors:
E-mail address: Aditya.bajracharya01@gmail.com (Aditya Bajracharya), projan.shakya@gmail.com (Projan Shakya).

their full potential to build robust and reliable AI models.

II. METHODOLOGY

A. Dataset Description

We used the Fertility dataset from the UCI Machine Learning Repository, designed to predict the fertility status of men based on various personal and lifestyle attributes. This dataset was collected as part of a study to understand the factors affecting male fertility. It consists of 100 instances, each characterized by 10 input features and an output class. The features include:

- Season (categorical)
- Age (integer)
- Childhood diseases (binary)
- Accident or serious trauma (binary)
- Surgical intervention (binary).
- High fever in the last year (categorical).
- Frequency of alcohol consumption (ordinal).
- Smoking habit (categorical).
- Hours spent sitting per day (integer).

The output class indicates the fertility status of the individual, categorized as either fertile or infertile. This dataset serves as an excellent resource for developing predictive models and conducting exploratory data analysis in medical and health-related research, particularly in understanding and identifying key factors influencing male fertility. Its well-defined features and clear output class make it suitable for various machine learning tasks, including classification and regression analysis.

B. Proposed Methodology

Principal Component Analysis (PCA) [2] is a statistical technique used to transform a dataset's features into a set of uncorrelated variables called principal components. These components are derived through an orthogonal transformation, where the first component captures the highest variance, the second component captures the second highest variance, and so on. In our study, the PCA algorithm was implemented from basic Python libraries to perform this transformation. The first step involves data preprocessing, where the dataset is loaded, and irrelevant attributes are removed. To encode the crop labels, a mapping dictionary is created, enabling numerical representation. Next, the dataset is converted into a matrix form, facilitating further computations. Exploratory Data Analysis is conducted through a pair plot, visualizing attribute relationships. Standardization is performed by subtracting attribute means, and the resulting standardized dataset is stored. The covariance matrix is computed, revealing attribute relationships. Eigenvalues and eigenvectors of the covariance matrix are calculated and sorted. By multiplying the standardized dataset with the sorted eigenvectors, a change of basis is achieved. The proportion of variance explained by each eigenvalue is determined. Finally, PCA is applied to project the transformed data onto different combinations of principal components, generating scatter plots and 3D visualizations.

C. Mathematical Formulae

Let's assume we have a matrix or dataset with dimensions

m×n.

The formula for the mean (μ) is:

$$\mu = \frac{\sum x_{ij}}{m\ x\ n} \qquad (1)$$

Where x_{ij} represents the element at the i-th row and j-th column of the matrix.

The formula for the covariance matrix Σ is:

$$\Sigma = \frac{1}{m-1}(X^T.X) \qquad (2)$$

For a square matrix A, the eigenvalues can be obtained by solving the characteristic equation:

$$|\,A - \lambda I| = 0 \qquad (3)$$

Where λ is the eigenvalue and I is the identity matrix of the same size as A. Given its eigenvalues $\lambda_1,\lambda_2,\ldots,\lambda_n$, the eigenvectors can be found by solving the equation:

$$(A - \lambda I)vi = 0 \qquad (4)$$

The proportion of variance (PoV) for each eigenvalue λ_i is given by:

$$PoV_i = \frac{\lambda_i}{\sum_{j=1}^{n} \lambda_j} \qquad (5)$$

D. Instrumentation Details

For performing the lab exercises, we used python as our go-to programming language. Several python libraries were included while performing the lab exercises. The main source of library that was used was Jupyter Notebook, which is a popular interactive coding environment where we can see the output and results instantly. Another library that we used was numpy, which is a library that is used to manipulate arrays and vectors that was used to perform mathematical calculations. Also, visualization was required for our lab exercises, so we opted to use matplotlib library which had many functions to represent graphs and plot them.

E. System Block Diagram

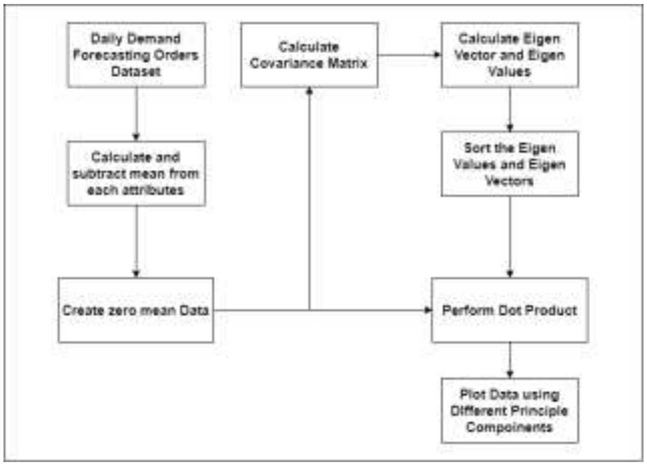
Daily Demand Forecasting Orders Dataset
Calculate Covariance Matrix
Calculate Eigen Vector and Eigen Values
Calculate and subtract mean from each attributes
Sort the Eigen Values and Eigen Vectors
Create zero mean Data
Perform Dot Product
Plot Data using Different Principle Compoinents

Figure 1: Block Diagram for Principal Component Analysis

The system block diagram is as shown in Figure 1: Block Diagram for Principal Component Analysis. First the dataset of Daily Demand of Forecasting Order was loaded into the program. The average of each feature of the dataset was loaded into panda library and it was subtracted from each of the features. This action created zero-mean data was created.

The covariance of zero-mean data was calculated and the eigen values and the eigen vectors of the covariance matrix was calculated. The covariance matrix was sorted and the topmost values of both the eigen value and eigen vector was Principal Component 1 and so on.

Different combinations of Principle Components were taken, and the data was plotted accordingly.

F. Eigen Values and Eigen Vectors

For a square matrix A, an Eigenvector and Eigenvalue make this equation true:

$$Av = \lambda v \qquad (6)$$

Where A is a m x m square matrix and λ is a scalar.

So, Eigenvalues and Eigenvectors are the scalar and vector quantities associated with Matrix used for linear transformation. The vector that does not change even after applying transformations is called the Eigenvector and the scalar value attached to Eigenvectors is called Eigenvalues. Eigenvectors are the vectors that are associated with a set of linear equations [3].

Eigenvalues are the scalar values associated with the eigenvectors in linear transformation. The word 'Eigen' is of German Origin which means 'characteristic'. Hence, these are the characteristic values that indicate the factor by which eigenvectors are stretched in their direction. It doesn't involve the change in the direction of the vector except when the eigenvalue is negative. When the eigenvalue is negative the direction is just reversed.

Eigenvectors for square matrices are defined as non-zero vector values which when multiplied by the square matrices give the scaler multiple of the vector, i.e. we define an eigenvector for matrix A to be "v" if it specifies the condition, Av = λv.

G. Change of Basis

The basis is a coordinate system used to describe vector spaces (sets of vectors). It is a reference that you use to associate numbers with geometric vectors.

To be considered as a basis, a set of vectors must:

- Be linearly independent
- Span the space

Every vector in space is a unique combination of the basis vectors. The dimension of a space is defined to be the size of a basis set. For instance, there are two basis vectors in $\mathbb{R}^2$ (corresponding to the x and y-axis in the Cartesian plane), or three in

$\mathbb{R}^3$.

Vectors can be represented as arrows going from the origin to a point in space. The coordinates of this point can be stored in a list. The geometric representation of a vector in the Cartesian plane implies that we take a reference: the directions given by the two axes x and y.

Basis vectors are the vectors corresponding to this reference. In the Cartesian plane, the basis vectors are orthogonal unit vectors (length of one), generally denoted as i and j [4].

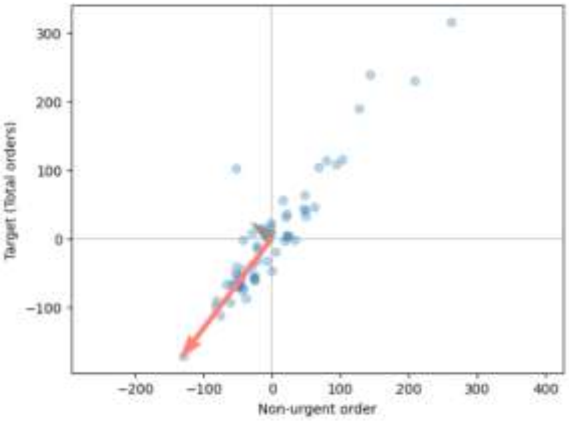

Figure 2: Change of Basis

H. Covariance Matrix

Covariance matrix is a type of matrix that is used to represent the covariance values between pairs of elements given in a random vector. The covariance matrix can also be referred to as the variance covariance matrix. This is because the variance of each element is represented along the main diagonal of the matrix.

A covariance matrix is always a square matrix. Furthermore, it is positive semi-definite, and symmetric. This matrix is very useful in stochastic modeling and principal component analysis.

Variance covariance matrix is defined as a square matrix where the diagonal elements represent the variance, and the off-diagonal elements represent the covariance. The covariance between two variables can be positive, negative, and zero. A positive covariance indicates that the two variables have a positive relationship whereas negative covariance shows that they have a negative relationship. If two elements do not vary together then they will display a zero covariance.

I. Proportion of Variance

When λᵢ are sorted in descending order, the proportion of variance explained by the r principal component is:

$$\frac{\sum_{i=1}^{r}\lambda_i}{\sum_{i=1}^{m}\lambda_i} = \frac{\lambda_1+\lambda_2+\cdots+\lambda_r}{\lambda_1+\lambda_2+\cdots+\lambda_m} \qquad (7)$$

If the dimensions are highly correlated, there will be small number of eigenvectors with large eigenvalues and r will be smaller than m but if dimensions are not correlated, r will be as large as m and PCA does not help.

III. Results

A. Problem 1

PCA on Random Data showed the effectiveness of Principal Component Analysis (PCA) in reducing the dimensionality of random data. Several 2D plots were generated to visualize the data before and after PCA.

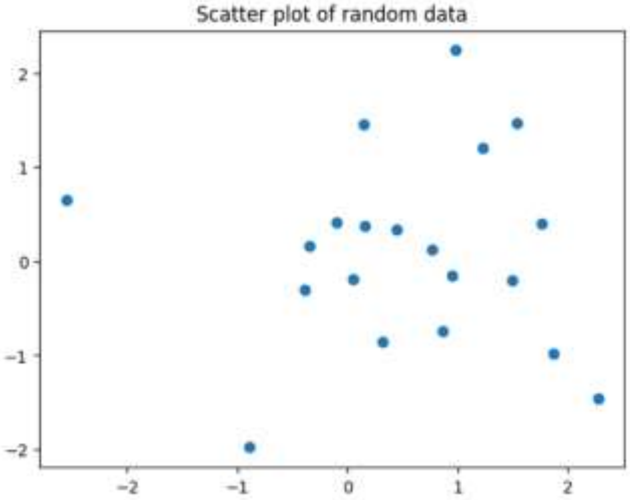

Figure 3: Scatterplot of Normal Data

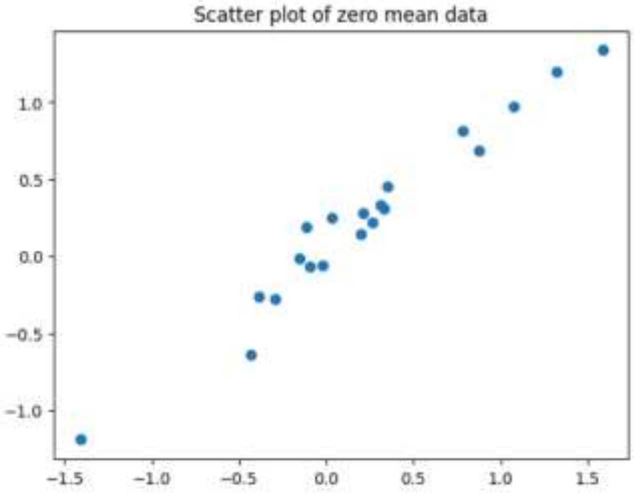

Figure 4: Scatterplot of Zero Mean Data

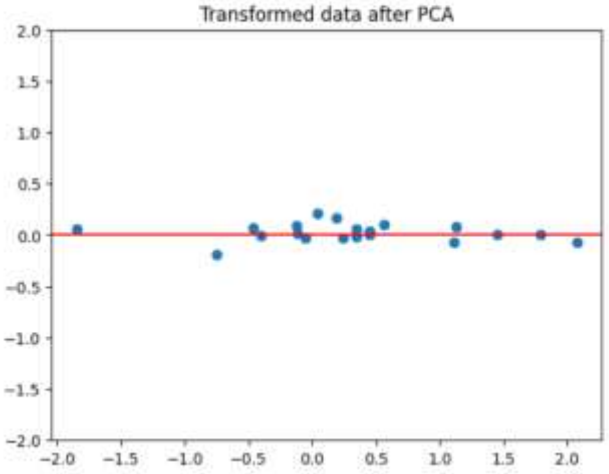

Figure 5: Data after Applying PCA

B. Problem 2

PCA on the Daily Demand Forecasting Orders demonstrated the efficiency of PCA in analyzing and visualizing complex datasets.

1. 2D plot of Principal Components

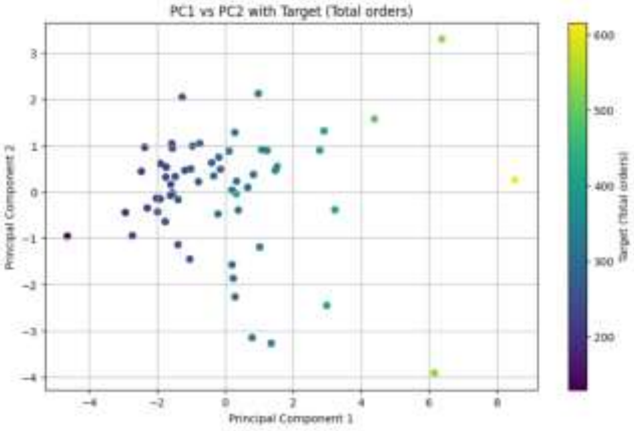

Figure 6: Plot using PC1 and PC2

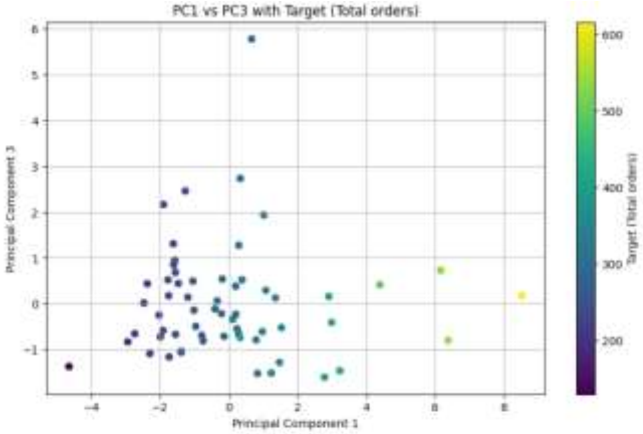

Figure 7: Plot using PC1 and PC3

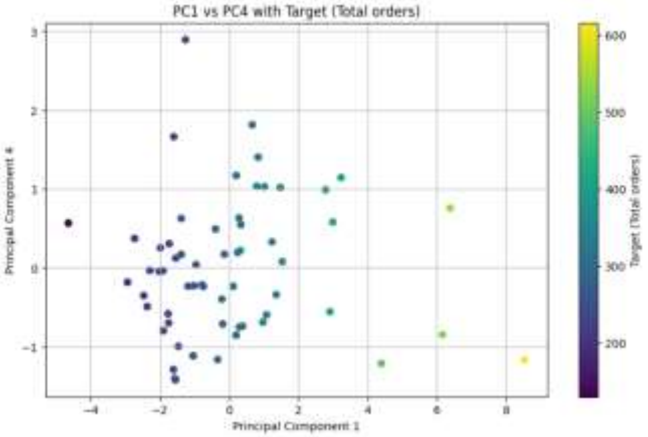

Figure 8: Plot using PC1 and PC4

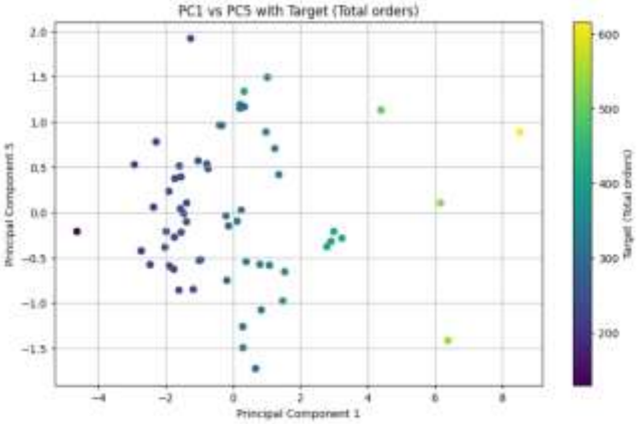

Figure 9: Plot using PC1 and PC5

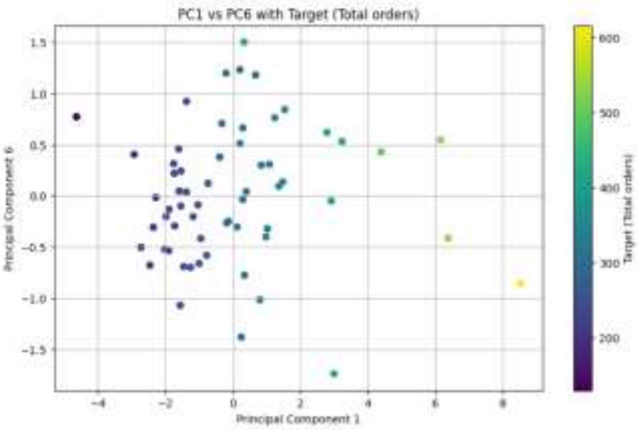

Figure 10: Plot using PC1 and PC6

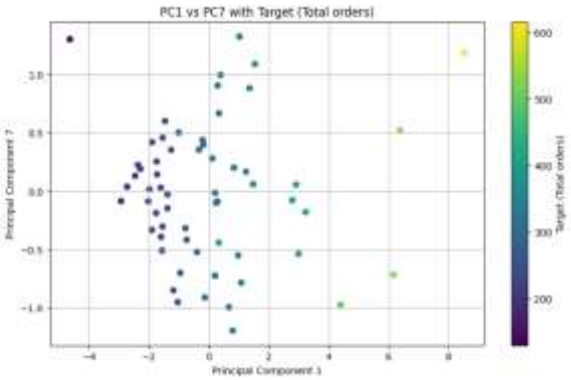

Figure 11: Plot using PC1 and PC7

2. 3D Plot for Principal Components

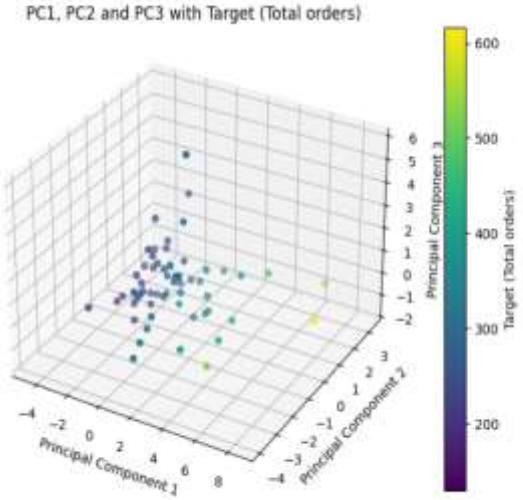

Figure 12: Plot using PC1, PC2 and PC3

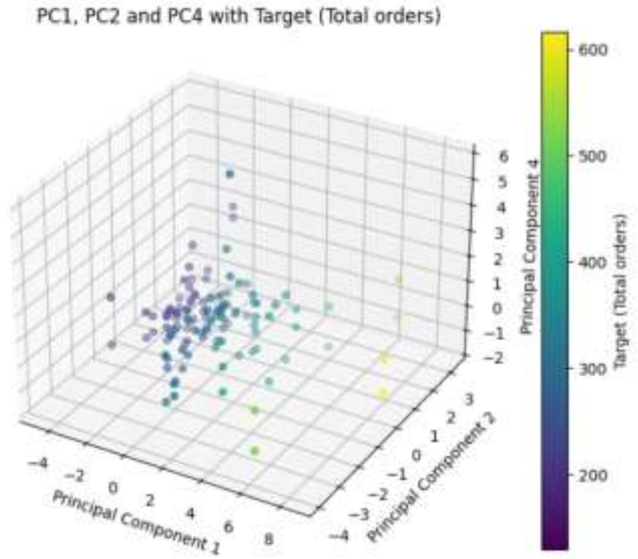

Figure 13: Plot using PC1, PC2 and PC4

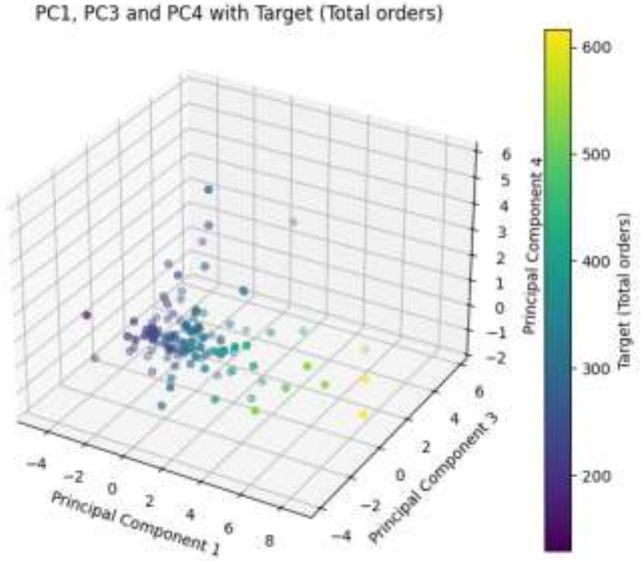

Figure 14: Plot using PC1, PC3 and PC4

3. Proportion of Variance due to Eigen Vectors

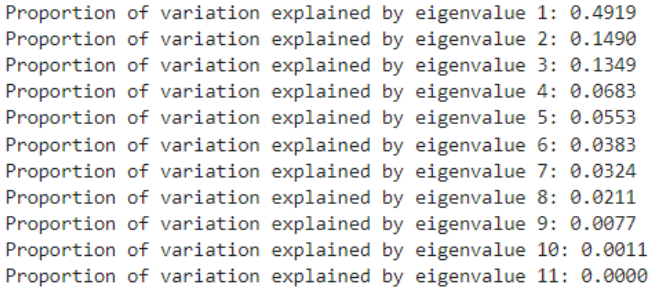
```
Proportion of variation explained by eigenvalue 1: 0.4919
Proportion of variation explained by eigenvalue 2: 0.1490
Proportion of variation explained by eigenvalue 3: 0.1349
Proportion of variation explained by eigenvalue 4: 0.0683
Proportion of variation explained by eigenvalue 5: 0.0553
Proportion of variation explained by eigenvalue 6: 0.0383
Proportion of variation explained by eigenvalue 7: 0.0324
Proportion of variation explained by eigenvalue 8: 0.0211
Proportion of variation explained by eigenvalue 9: 0.0077
Proportion of variation explained by eigenvalue 10: 0.0011
Proportion of variation explained by eigenvalue 11: 0.0000
```

Figure 15: Proportion of Variance Due to Eigen Vectors

The proportion of variance is calculated by dividing the corresponding eigenvalue to the number of variables used and the proportion of variance due to different eigenvalues is shown in figure.

IV. Discussion and Analysis

In Problem 1 initially, the random data points were scattered across the plot with no apparent structure. However, after applying PCA, the data points were transformed into a new coordinate system aligned with the principal components.

In Problem 2 initially, the dataset contained multiple features, making it difficult to visualize patterns. By applying PCA, the data was transformed into a lower-dimensional space, preserving the most significant variations. This transformation allowed for more accessible visualization. The 2D plots showed the distribution of the data points in reduced dimensions, revealing clusters and patterns that were not apparent in the original dataset. The 3D plots provided an even more comprehensive representation of the data, allowing for the examination of relationships between three principal components.

V. Conclusion

Hence, from this lab exercise, we learned about the working of the PCA algorithm and visualized the different combinations of Principal Component axes with their data visualization.

In Problem 1, PCA was able to capture the underlying structure of the randomly generated data, showcasing its ability to reveal patterns in synthetic datasets. In Problem 2, PCA successfully helped to visualize the total sales target of the store with the help of a different combination of Principal Components while preserving its information.

The practical implementations confirmed that Principal Component Analysis (PCA) is a valuable tool for feature extraction, providing insights into the intrinsic structure of data and reducing its dimensionality. Despite its limitations, such as linearity assumptions and sensitivity to outliers, PCA showcased its potential in various domains including data analysis, and pattern recognition. Future research focusing on alternative feature extraction techniques and larger datasets can further enhance our understanding and applicability of PCA, contributing to our overall learning experience in the field.

VI. References

[1] J. Gil, L. Girela, J. De Juan, M. Gomez-Torres, and E. M. John, Fertility, UC Irvine Machine Learning Repository, 2013.

[2] S. Sehgal, H. Singh, M. Agarwal, Shantanu and V. Bhasker, "Data analysis using principal component analysis," *2014 International Conference on Medical Imaging, m-Health and Emerging Communication Systems (MedCom),* 2014.

[3] "Geeks for Geeks," Geeks for Geeks, 01 03 2024. [Online]. Available: https://www.geeksforgeeks.org/eigen-values/.

[4] H. Jean, "Towards Data Science," Towards Data Science, 01 February 2021. [Online]. Available: https://towardsdatascience.com/essential-math-for-data-science-basis-and-change-of-basis-f7af2348d463.

Aditya Bajracharya

Aditya Bajracharya is a student of Computer Engineering in Thapathali Campus, IOE. Currently pursuing his Bachelor’s degree in Computer Engineering from Tribhuvan University (Kathmandu, Nepal). Currently a student in the Thapathali Campus, Aditya’s interest remains in the field of Artificial Intelligence.

Projan Shakya

Projan Shakya is a student of Computer Engineering in Thapathali Campus, IOE. Currently pursuing his Bachelor’s degree in Computer

Engineering from Tribhuvan University (Kathmandu, Nepal). Currently a student in the Thapathali Campus, Projan’s interest remains in the field of Artificial Intelligence.