D212 Data Mining II

OFM4 Dimensionality Reduction Methods (Task 2)

Performance Assessment

Hillary Osei (Student ID #011039266)

Western Governors University, College of Information Technology

Program Mentor: Dan Estes

August 20, 2025

Table of Contents

Part I: Research Question

Section A1. Proposal of Question

The research question proposed is “What are the patterns in patient characteristics that can help hospitals better manage patient admissions.” The medical dataset (“medical_cleaned.csv”) is used to address this.

Section A2. Defined Goal

The goal of this analysis is to reduce the number of variables in the dataset and identify important factors that define patient profiles, which can be used to understand patterns concerning hospital admissions.

Part II: Method Justification

Section B1. Explanation of PCA

Principal Component Analysis (PCA) is used to capture variations in a dataset and reduce dimensions. It works by calculating the covariance matrix, a matrix that shows the variance between two variables (Cuemath, n.d.), then it calculates the matrix’s eigenvectors. The eigenvector with the highest eigenvalue identifies the first principal component because it determines the direction of the highest variance. Then, the eigenvector with the second highest eigenvalue determines the second principal component and so forth (Jain, 2025). The expected outcome is a simpler dataset, meaning a lesser number of variables in the dataset, while key trends in the data are retained (Jaadi, 2025).

Section B2. PCA Assumption

According to Keebola, a key assumption for PCA is that “...there is a linear relationship between the features.” (Keebola, 2022) This is because PCA is based on Pearson correlation coefficients, which can only measure linear relationships between variables (Laerd Statistics, n.d.)

Part III: Data Preparation

Section C1. Continuous Data Set Variables

The continuous variables used to address the research question are:

- Population

- Children
- Age
- Income
- VitD_levels
- Doc_visits
- Full_meals_eaten
- vitD_supp
- Initial_days
- Total Charge
- Additional_charges

Section C2. Standardization of Data Set Variables

To prepare for PCA, the continuous variables in the dataset were standardized using StandardScaler from the scikit-learn library. This transforms each variable so that it has a mean of 0 and a standard deviation of 1 (GeeksforGeeks, 2025). The scaler was first fit to the dataset to calculate the mean and standard deviation of each column. Then, the standardized variables were stored in a new DataFrame("scaled_data").

```
[15]: # Use Standard Scaler to standardize data
      from sklearn.preprocessing import StandardScaler, RobustScaler
      sc = StandardScaler()
      sc.fit(df)
      scaled_data_array = sc.transform(df)
      scaled_data = pd.DataFrame(scaled_data_array, columns = df.columns)
      scaled_data.head()
```

[15]:

	Population	Children	Age	Income	VitD_levels	Doc_visits	Full_meals_eaten	vitD_supp	Initial_days	TotalCharge	Additional_charges
0	-0.473168	-0.507129	-0.024795	1.615914	0.583603	0.944647	-0.993387	-0.634713	-0.907310	-0.727185	0.765005
1	0.090242	0.417277	-0.121706	0.221443	0.483901	-0.967981	0.990609	0.956445	-0.734595	-0.513228	0.715114
2	0.482983	0.417277	-0.024795	-0.915870	0.046227	-0.967981	-0.001389	-0.634713	-1.128292	-1.319983	0.698635
3	-0.526393	-0.969332	1.186592	-0.026263	-0.687811	-0.967981	-0.001389	-0.634713	-1.244503	-1.460517	0.009004
4	-0.315586	-0.507129	-1.526914	-1.377325	-0.260366	-0.011667	-0.993387	2.547602	-1.261991	-1.467285	-1.408991

The cleaned/scaled dataset is attached, titled "D212_scaled_task2.csv."

Part IV: Analysis

Section D1. Principal Components

PCA was performed on the standardized dataset using the PCA() function. After fitting the PCA model, the components_ attribute was used to extract the principal component matrix. The matrix shows the loadings of each original standardized variable on each principal component.

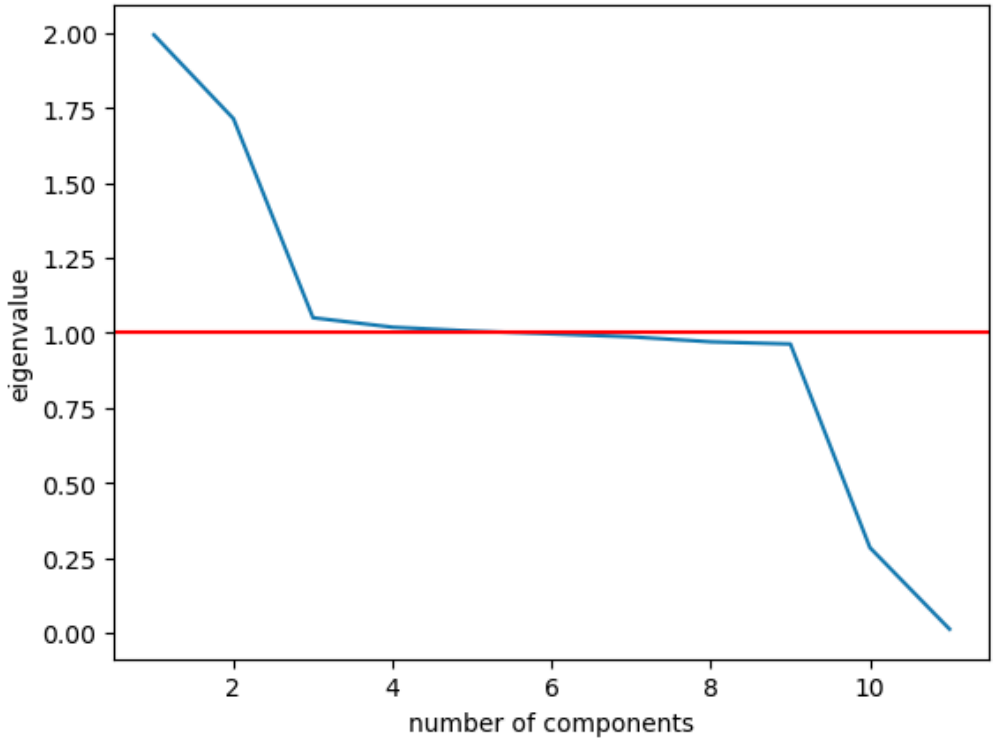
Each row in the matrix represents a principal component, and each value within the row indicates a variable's contribution to the component (Harvey & Hanson, 2024). In total, 11 principal components were generated to match the 11 original continuous variables.

```
[[ 9.99001014e-01  2.60352278e-03 -2.37958539e-02  9.11767614e-03
   2.48169228e-03  7.44680929e-03 -1.48903257e-02  8.02231064e-03
   1.28685060e-02  1.38269581e-02 -2.52258616e-02]
 [ 3.34835363e-02  2.24188565e-02  5.13345102e-01 -3.29468046e-02
   1.81835086e-02  6.26060252e-03  1.12513865e-02  1.24465238e-02
   1.58592141e-02  2.63462393e-02  8.55669666e-01]
 [-7.37424702e-03  2.39168801e-02  1.55482390e-02  9.96299452e-01
  -5.99876884e-02  1.41436681e-02 -1.08473077e-02  2.77725625e-03
  -2.75711006e-02 -2.91643388e-02  3.13781642e-02]
 [ 3.35333528e-03 -1.67614221e-01  8.27523055e-03 -5.57056477e-02
  -9.83181218e-01 -1.41310655e-02 -2.89911360e-02  2.55400552e-02
   6.22541098e-03  5.12464603e-03  1.78839797e-02]
 [-1.12163528e-02  8.88859035e-01 -1.65206575e-02 -1.18188528e-02
  -1.47219547e-01 -1.01049334e-02 -9.51683610e-03  1.29601700e-02
   3.00624389e-01  3.10383051e-01 -2.53828422e-02]
 [ 1.79552996e-02  4.24472628e-01  6.35793817e-03 -5.11696273e-02
  -7.89671225e-02  4.67868769e-03  2.30313054e-02 -7.13279685e-02
  -6.24763565e-01 -6.43719698e-01  1.61698164e-02]
 [-7.79200967e-03  2.30500902e-02 -4.11584218e-03 -5.19863882e-03
   2.23592446e-02  1.63407339e-02 -4.01798139e-02  9.95892566e-01
  -4.98507273e-02 -5.09286162e-02 -1.00904677e-02]
 [ 8.03406030e-03 -3.93218298e-03  2.50554366e-03  1.48949091e-02
   1.36952793e-02 -9.98963679e-01  3.41381871e-02  1.70914026e-02
  -5.82959915e-03 -7.34477197e-03  5.51368522e-03]
 [ 1.33078808e-02 -5.11540588e-03 -2.10858870e-02  1.00260737e-02
  -2.84929068e-02  3.44245695e-02  9.97492878e-01  4.20688545e-02
   1.38555133e-02  1.72801071e-02 -1.51397119e-03]
 [ 7.71161821e-03  1.82630073e-03  8.56938194e-01  2.76100697e-03
  -3.10161484e-03 -2.00450266e-05  1.72620450e-02 -9.70755868e-04
   2.01183005e-02 -2.16729880e-02 -5.14201807e-01]
 [-4.14887057e-04 -6.36051206e-04  2.18192470e-02  7.82638088e-04
  -1.00349803e-03 -1.05648873e-03 -1.63412136e-03 -4.73708039e-04
  -7.17609664e-01  6.95779557e-01 -2.10924535e-02]]
```

Section D2. Identification of The Total Number of Components

To determine how many principal components to retain, the Kaiser Criterion was applied. This rule states to only keep components with eigenvalues greater than 1 (Steiger, 2015). After performing the PCA, the eigenvalues of all 11 components were calculated and visualized using a scree plot. A horizontal line at eigenvalue = 1 was used to identify the cut-off. Based on the criterion, 5 principal components had eigenvalues greater than 1 and were therefore retained.

The scree plot below shows a drop in eigenvalues after the fifth component, further supporting the decision to retain 5 components.




```
# Sort eigenvalues in descending order
eigenvalues_sorted = np.sort(eigenvalues)[::-1]

# Calculate the number of eigenvalues greater than 1
eigen_components = np.sum(eigenvalues_sorted > 1)

print(f"Number of Principal Components to Retain: {eigen_components}")
```

```
Number of Principal Components to Retain: 5
```

Section D3. Variance of Each Component

To determine the variance for each of the retained principal components, the PCA model was re-fitted using the 5 components. The "explained_variance_ratio_" attribute was used to calculate the proportion of variance that each component contributes. Results show that the first principal components explained 18.14% of the variance, the second explained 15.59%, the third 9.55%, the fourth 9.27%, and the fifth 9.15%.

```
# Get variance per retained principal component

retained_pc = PCA(n_components=5, random_state=42)
retained_pc.fit(scaled_data)
variance_pca = retained_pc.transform(scaled_data)

captured_variance = retained_pc.explained_variance_ratio_ * 100

# Print the captured variance for each component
for i, var in enumerate(captured_variance):
    print(f"Principal Component {i+1}: {var:.2f}%")
```

```
Principal Component 1: 18.14%
Principal Component 2: 15.59%
Principal Component 3: 9.55%
Principal Component 4: 9.27%
Principal Component 5: 9.15%
```

Section D4. Total Variance Captured by Components

To determine the total variance captured by the top five principal components, the cumulative explained variance was calculated using the np.cumsum() function. The result was a running total of variance explained as each component was added. The analysis showed that the first five components in total captured 61.70% of the total variance in the dataset.

```
# Get total variance
cumulative_variance = np.cumsum(captured_variance[:5])
print(f"Total variance explained by top 5 components: {cumulative_variance[-1]:.2f}%\n")

print("Cumulative variance up to 1st principal component =", cumulative_variance[0])
print("Cumulative variance up to 2nd principal component =", cumulative_variance[1])
print("Cumulative variance up to 3rd principal component =", cumulative_variance[2])
print("Cumulative variance up to 4th principal component =", cumulative_variance[3])
print("Cumulative variance up to 5th principal component =", cumulative_variance[4])
```

```
Total variance explained by top 5 components: 61.70%

Cumulative variance up to 1st principal component = 18.136305395664248
Cumulative variance up to 2nd principal component = 33.729456264420165
Cumulative variance up to 3rd principal component = 43.282392299365114
Cumulative variance up to 4th principal component = 52.549343967646195
Cumulative variance up to 5th principal component = 61.698580376651606
```

Section D5. Summary of Data Analysis

PCA was applied to a dataset containing 11 continuous variables to reduce dimensionality and identify key patterns. After standardizing the data, PCA was conducted, and the Kaiser Criterion was used to determine the number of components to retain. Five principal components were selected since each had an eigenvalue greater than 1. Cumulatively, the five components

captured a total of 61.70% of the variance in the original dataset, meaning that the components capture most of the meaningful information in the dataset. The individual variance explained by each component ranged from 9.15% (from the fifth component) to 18.14% (from the first component). The results of the analysis suggest that patient characteristics can be summarized using just five key components instead of the 11 original variables. Thus, the hospital can focus on a much smaller number of factors (like age, income, number of doctor's visits, etc) when analyzing patient trends. This can potentially improve decision-making concerning patient admissions and allocation of resources.

Part V: Attachments

Section E. Sources for Third-Party Code

Middleton, K. (n.d.). *Welcome to getting started with principal component analysis (PCA) [Slide 11]*. Western Governors University.

Middleton, K. (n.d.). *Welcome to getting started with principal component analysis (PCA) [Slide 12]*. Western Governors University.

Middleton, K. (n.d.). *Welcome to getting started with principal component analysis (PCA) [Slide 13]*. Western Governors University

Middleton, K. (n.d.). *Welcome to getting started with principal component analysis (PCA) [Slide 15]*. Western Governors University.

Middleton, K. (n.d.). *Welcome to getting started with principal component analysis (PCA) [Slide 16]*. Western Governors University.

Section F. Sources

Cuemath. (n.d.). *Covariance Matrix - Formula, Examples, Definition, Properties*. Cuemath. Retrieved August 7, 2025, from https://www.cuemath.com/algebra/covariance-matrix/

Harvey, D. T., & Hanson, B. A. (2024, April 26). *Understanding Scores and Loadings*. LearnPCA. Understanding Scores and Loadings

Jaadi, Z. (2025, June 23). *Principal Component Analysis (PCA): Explained Step-by-Step*. Built In. Retrieved July 17, 2025, from https://builtin.com/data-science/step-step-explanation-principal-component-analysis

Jain, S. (2025, July 12). *StandardScaler, MinMaxScaler and RobustScaler techniques - ML*. GeeksforGeeks. Retrieved July 17, 2025, from https://www.geeksforgeeks.org/machine-learning/standardscaler-minmaxscaler-and-robustscaler-techniques-ml/

Jain, S. (2025, July 23). *Mathematical Approach to PCA*. GeeksforGeeks. Retrieved August 7, 2025, from https://www.geeksforgeeks.org/machine-learning/mathematical-approach-to-pca/

Keebola. (2022, April 2). *A Guide to Principal Component Analysis (PCA) for Machine Learning*. Keboola. Retrieved July 17, 2025, from https://www.keboola.com/blog/pca-machine-learning

Laerd Statistics. (n.d.). *How to perform a principal components analysis (PCA) in SPSS Statistics*. Laerd Statistics. Retrieved August 7, 2025, from https://statistics.laerd.com/spss-tutorials/principal-components-analysis-pca-using-spss-statistics.php

Steiger, J. H. (2015, February 16). *Principal Components Analysis*. Statpower. https://statpower.net/Content/312/R%20Stuff/PCA.html