**D212 Data Mining II**

**Task 2: Dimensionality Reduction**

Petra I Bier

Western Governors University

D212 Data Mining II

December 1, 2024

Dr. Kesselly Kamara PhD

**D212 Data Mining II**

**Task 2: Dimensionality Reduction**

**Part I: Question**

A :

A1: Question

What patient characteristics have similar traits that can be used to identify important factors that may affect hospital care improvement?

A2: Goal

As medical costs rise for patients and insurance companies, thorough data analysis to discover patterns and trends that can reduce costs and prevent readmissions would benefit all parties. Based on the analysis results, preventative interventions could be developed to reduce insurance company costs, reduce the length of stay, and avoid hospital readmissions. This Principal Component Analysis aims to reduce the dimensions of the variables to determine the most critical patient characteristics.

**Part II**

B :

B1: Principal Component Analysis

Working with large data sets can make it challenging to pick out which features could be used in an analysis due to the large volume of variables. Principal Component Analysis (PCA) can evaluate the features to see which ones are the most interesting, where "interesting is measured by the amount that the observations vary" (James et al., 2023, p. 505). The PCA algorithm creates a new set of variables called principal components (PC), each a linear combination of all the features used in the analysis.

The first PC created has the most significant weighted variance, the second has the second most variance, and so on. These PCs can be visualized as vectors, with each vector plotted orthogonally to the previous vector. This means that each direction of a PC in a plot represents the direction and size of the particular feature variability within a PC. Or, simply put, how “interesting” the PC is.

The next step in PCA dimension reduction is to create a loading matrix. The algorithm automatically orders the PCs from highest variability to lowest. The matrix allows for a numeric evaluation of the weights within each PC, also known as loadings, which show the strength of each variable’s contribution to the PC. The loadings are on a scale of -1 to 1, with -1 having high variability with a negative correlation and +1 being highly variable in a positive direction. Values at or close to zero have little variability. These values help interpret the meaning of each PC by showing which variables are most influential. A heat map can be created to simplify the interpretation of each variable on each PC.

Once the matrix is created and evaluated, the Eigenvalues, also called explained variance, can be extracted from the PCA model. When using standardized data, these values are on a scale of 0 up to the number of total variables, so the total of all Eigenvalues must equal the number of variables. The larger numbers mean that the PC has more variability and can be considered more interesting. A value of less than one implies that the PC only has as much variability as any single value.

The explained variance ratio represents this value in a percentage, making it easier to interpret by allowing a person to see how much the PC contributes to the total variance. This value can be summed up from all the PCs that have been kept, leading to the understanding of how interesting all the kept PCs are.

The number of PCs to keep can be decided in several different ways. One is to have a predetermined sum of the explained variance ratio. Another can be determining the importance of each PC by using the Kaiser criterion, which was used for this analysis. Since the number of PCs chosen is less than the initial number of variables, the expected outcome of having data with reduced dimensionality has been met.

B2: Method Assumption

When working with statistical models, assumptions regarding the type of data being worked with must be met. One such assumption For PCA is that the variables are continuous(Lund Research Ltd, 2018). The statistical computations calculate weighted sums of the original variables. These mathematical functions only give meaningful results using continuous values (Firmin, 2018). This also explains why some ordinal data, like a Likert scale, can be used within a PCA.

**Part III Preparation**

C :

C1: Variable Description

PCA requires that all the data should be continuous numeric. An exception was made for the survey data, which is ordinal data on a 7-point Likert scale and can be treated as continuous (Skoczylis, 2022). The data set was examined and passed down through several cleaning steps. The chart below shows each variable's data type (Soetewey, 2019) in the data to determine if it would be selected for the final model. The final variables chosen for the analysis are highlighted in blue.

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Variable Type | Data Type | Selected final analysis |
| City | Categorical | Nominal | No |
| State | Categorical | Nominal | No |
| County | Categorical | Nominal | No |
| Zip | Numeric | Discrete | No |
| Lat | Numeric | Continuous | Yes |
| Lng | Numeric | Continuous | Yes |
| Population | Numeric | Discrete | No |
| Area | Categorical | Nominal | No |
| Timezone | Categorical | Nominal | No |
| Job | Categorical | Nominal | No |
| Children | Numeric | Discrete | No |
| Age | Numeric | Continuous | Yes |
| Education | Categorical | Nominal | No |
| Employment | Categorical | Nominal | No |
| Income | Numeric | Continuous | Yes |
| Marital | Categorical | Nominal | No |
| Gender | Categorical | Nominal | No |
| ReAdmis | Categorical | Nominal / Bivariate | No |
| VitD\_levels | Numeric | Continuous | Yes |
| Doc\_visits | Numeric | Discrete | No |
| Full\_meals\_eaten | Numeric | Discrete | No |
| VitD\_supp | Numeric | Discrete | No |
| Soft\_drink | Categorical | Nominal | No |
| Initial\_admin | Categorical | Nominal | No |
| HighBlood | Categorical | Nominal / Bivariate | No |
| Stroke | Categorical | Nominal / Bivariate | No |
| Complication\_risk | Categorical | Nominal | No |
| Overweight | Categorical | Nominal / Bivariate | No |
| Arthritis | Categorical | Nominal / Bivariate | No |
| Diabetes | Categorical | Nominal / Bivariate | No |
| Hyperlipidemia | Categorical | Nominal / Bivariate | No |
| Backpain | Categorical | Nominal / Bivariate | No |
| Anxiety | Categorical | Nominal / Bivariate | No |
| Allergic rhinitis | Categorical | Nominal / Bivariate | No |
| Reflux esophagitis | Categorical | Nominal / Bivariate | No |
| Asthma | Categorical | Nominal / Bivariate | No |
| Services | Categorical | Nominal | No |
| Initial\_days | Numeric | Continuous | Yes |
| TotalCharge | Numeric | Continuous | Yes |
| Additional\_charges | Numeric | Continuous | Yes |
| Item1 | Numeric | Ordinal | Yes |
| Item2 | Numeric | Ordinal | Yes |
| Item3 | Numeric | Ordinal | Yes |
| Item4 | Numeric | Ordinal | Yes |
| Item5 | Numeric | Ordinal | Yes |
| Item6 | Numeric | Ordinal | Yes |
| Item7 | Numeric | Ordinal | Yes |
| Item8 | Numeric | Ordinal | Yes |

C2: Variable Standardization

When running a PCA analysis, it is necessary to preprocess the data. Preprocessing tasks can include cleaning and ensuring no missing or null values. Data wrangling tasks can also be part of the preprocessing. These tasks ensure an accurate, well-fitted model. The most crucial preprocessing task performed on this data was normalization, which allowed for comparing the different measurement units in the patient variables.

When large distances between variables occur due to being on different scales, the larger measurements will dominate the influence. Normalizations of data allow variables on different measurement scales to be compared using the same comparison scale. A PCA algorithm would give unequal weight to these variables without normalization, as the variables with larger absolute values would dominate the algorithm's variance calculation. By scaling the variables, each feature contributes equally to the calculations during the clustering process (Kumar, 2024).

The data was normalized using StandardScaler() from sklearn.preprocessing library (Gramfort et al., 2022).

scaler = StandardScaler()

df\_norm = scaler.fit\_transform(df\_pca)

df\_norm = pd.DataFrame(df\_norm)

The difference in the numerical values can be seen in a comparison before and after the scaling.

A screenshot of a graph

Description automatically generated

Figure 1: Before Normalization

A screenshot of a graph

Description automatically generated

Figure 2: After Normalization

A file of the cleaned and normalized data set is attached as a .csv file labeled PCA\_scaled.csv.

**Part IV Analysis**

D :

D1: PCA Matrix

The normalized data set with 16 variables was fit to the PCA model, and a matrix was returned with all 16 variables.

pca = PCA()

pca\_med = pca.fit\_transform(df\_norm)

pca\_cols = [f'PC{i+1}' for i in range(pca.n\_components\_)]

loadings = pd.DataFrame(

pca.components\_.T,

columns= pca\_cols,

index= df\_pca.columns

)

A table of numbers with a white background

Description automatically generated

A heat map was also made to visualize all the relationships in the component matrix (Lee, 2018).

plt.figure(figsize=(12,10))

sns.heatmap(loadings,

annot=True,

cmap='coolwarm',

center=0,

fmt='.2f',

annot\_kws={'size': 8})

plt.title('PCA Loadings Heatmap')

plt.tight\_layout()

A screenshot of a graph

Description automatically generated

D2: Component Choice

The total number of components chosen was based on the Kaiser criterion results. All eigenvalues that are one or greater are kept and can be determined using a scree plot. A value of one or greater means that the PC has contributed one column’s variability to a PC. If the eigenvalue is less than one, then the PC contributes less to variability than a single column and is not included in the analysis (Larose & Larose, 2019, p. 176). The Eigenvalues can also be verified numerically (Lee, 2018), as shown in section D3. Based on these criteria, six PCs were kept for the analysis.

ex\_var = pca.explained\_variance\_

ex\_var

A number on a white background

Description automatically generated

plt.figure(figsize=(13, 6))

plt.plot(range(1, len(pca.explained\_variance\_) + 1), # x-axis: component numbers

pca.explained\_variance\_, # y-axis: eigenvalues

'bo-', # blue line with circle markers

linewidth=2)

plt.title("Kaiser Criterion", fontsize=16)

plt.xlabel("Principal Component", fontsize=16)

plt.ylabel("Eigenvalue", fontsize=16)

plt.axhline(y=1, color='r', linestyle='--', label='Eigenvalue = 1')

plt.grid()

plt.legend()

plt.show()

A graph with a line going up

Description automatically generated

D3: Principal Component Variance

The explained variance was identified (Lee, 2018). The explained variance ratio was also calculated to ease the interpretation of the weight of each PC.

eigenvalues = pca.explained\_variance\_

kaiser\_criterion = eigenvalues > 1

n\_components\_kaiser = sum(kaiser\_criterion)

print(f"Number of components with eigenvalue > 1: {n\_components\_kaiser}")

important\_components = np.arange(1, n\_components\_kaiser + 1)

print(components to keep according to the Kaiser Criterion: {important\_components}")

for i, (ev, satisfy) in enumerate(zip(eigenvalues, kaiser\_criterion), start=1):

if satisfy:

print(f"Component {i} explained variance: {ev}")

A black text with numbers

Description automatically generated with medium confidence

#Explained variance ratio

explained\_variance\_ratio = pca.explained\_variance\_ratio\_

for i in range(6):

print(f"Component {i+1} explained variance ratio: {round(explained\_variance\_ratio[i]\*100, 2)}%")

A black text on a white background

Description automatically generated

D4: PC Total Variance

The total variance was calculated by summing the explained variance of the final six PCs used in the analysis. The total variance ratio was also calculated using the sum of the explained variance ratio (Pedregosa et al., 2011), as this can be easier to interpret than the sum of the Eigenvalues since it is a percentage value.

total\_variance = sum(ev for ev in eigenvalues[:6] if ev > 1)

print(f"Total explained variance in kept Principal Components: {round(total\_variance, 2)}")



#Total variance ratio of the 6 components

tot\_variance = np.sum(pca.explained\_variance\_ratio\_[kaiser\_criterion])

print(

f"Total variance ratio explained by kept components: {round(tot\_variance \*100, 2)}%"

)



D5: PCA Analysis

After the PCA algorithm was run and the PC was chosen, the loadings were plotted to visualize the weights of the components and the relationships of the variables within the components (Bruce et al., 2019). A threshold of 0.4 was chosen to benchmark the fundamental variables within each component (Chouinard, 2023).

threshold = 0.4

maxPC = 1.01 \* np.abs(loadings.values).max()

f, axes = plt.subplots(6, 1, figsize=(5, 5), sharex=True)

for i, ax in enumerate(axes):

pc\_loadings = loadings[f'PC{i+1}']

color = ["blue" if l > 0 else "red" for l in pc\_loadings]

ax.axhline(color="#888888") # zero line

ax.axhline(y=threshold, color="grey", linestyle="--") # upper threshold

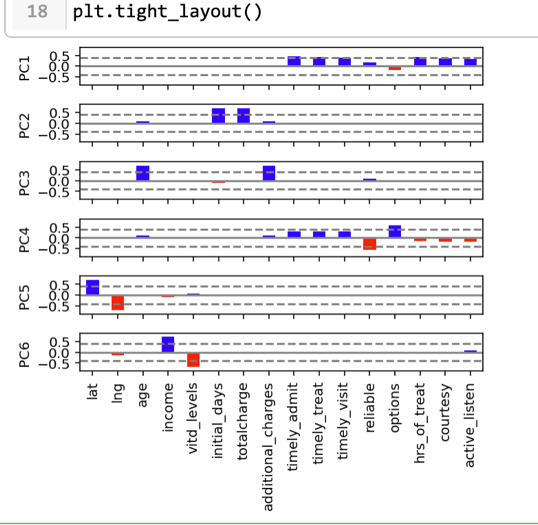
ax.axhline(y=-threshold, color="grey", linestyle="--") # lower threshold

pc\_loadings.plot.bar(ax=ax, color=color)

ax.set\_ylabel(f"PC{i+1}")

ax.set\_ylim(-maxPC, maxPC)

plt.tight\_layout()



The features were then shown for each component so that the feature variability relationships could be analyzed for each PC.

important\_features = {}

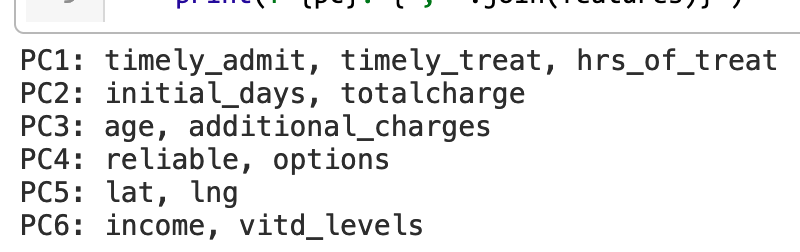
for column in loadings.columns[:6]:

important\_features[column] = loadings.index[loadings[column].abs() >

threshold].tolist()

for pc, features in important\_features.items():

print(f"{pc}: {', '.join(features)}")



The components can be summarized as follows:

PC1: Patient Care Experience

* 18.47% of variance
* Positive correlation:
  + Timely admission (0.455)
  + Timely treatment (0.428)
  + Hours of treatment (0.410)
* Signifies overall patient satisfaction with timing aspects of care delivery

PC2: Cost and Duration

* 12.45% of variance
* Positive correlation:
  + Total charge (0.701)
  + Length of stay (0.700)
* Signifies a correlation between length of stay and total charges

PC3: Age and Additional Charges

* 10.72% of variance
* Positive correlation:
  + Additional charges (0.694)
  + Age (0.694)
* Signifies older patients have more additional charges

PC4: Service Trade-offs

* 10.31% of variance
* Negative correlation:
  + Options (0.577)
  + Reliability (-0.553)
* Signifies an inverse relationship between customer-perceived service reliability and treatment options

PC5: Geographic Distribution

* 6.96% of variance
* Negative correlation:
  + Latitude (0.706)
  + Longitude (-0.699)
* Signifies geographic variation in services

PC6: Socioeconomic Health Relationship

* 6.34% of variance
* Negative correlation:
  + Income (0.718)
  + Vitamin D levels (-0.674)
* Signifies an inverse relationship between income and vitamin D levels

**Part VI**

E: Web sources

Chouinard, J. C. (2023, December 5). *What are PCA Loadings (with Python Example)*? JC Chouinard. Retrieved November 20, 2024, from <https://www.jcchouinard.com/pca-loadings/>

Firmin, S. (2019, July 29). *Tidying up with PCA: An Introduction to Principal Components Analysis*. Medium- Towards Data Science. Retrieved November 20, 2024, from <https://towardsdatascience.com/tidying-up-with-pca-an-introduction-to-principal-components-analysis-f876599af383?gi=40e97fbee3ee>

Gramfort, A., Blondel, M., & Grisel, O. (2022, October 18). StandardScaler. Scikit Learn. Retrieved November 15, 2024, from <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html>

Kumar, D. (2018, December 25). Introduction to Data Preprocessing in Machine Learning. Towards Data Science. Retrieved November 15, 2024, from <https://futuremachinelearning.org/understanding-silhouette-score-a-key-metric-for-clustering/>

Lee, E., PhD (2004, April 8). *Secrets of PCA: A Comprehensive Guide to Principal Component Analysis with Python and Colab*. Medium. Retrieved November 20, 2024, from <https://drlee.io/secrets-of-pca-a-comprehensive-guide-to-principal-component-analysis-with-python-and-colab-6f7f3142e721>

Lund Research Ltd. (2018). *Principal Components Analysis (PCA) using SPSS Statistics*. Laerd Statistics. Retrieved November 20, 2024, from <https://statistics.laerd.com/spss-tutorials/principal-components-analysis-pca-using-spss-statistics.php>

Skoczylis, J. (2022). *Reducing Data*. Stats Made Easy. Retrieved November 20, 24, from <https://www.statsmadeasy.com/stats-concepts/7-reducing-data>

Soetewey, A. (2019, December 30). Variable types and examples. Stats and R. Retrieved August 26, 2023, from <https://statsandr.com/blog/variable-types-and-examples/>

F: Sources

Bruce, P., Bruce, A., & Gedeck, P. (2019). Practical Statistics for Data Scientists: 50+ Essential Concepts Using R and Python (2nd ed.). O'Reilly Media.

James, G., Witten, D., Hastie, T., Tibsharani, R., & Taylor, J. (2023). *An Introduction to Statistical Learning with Applications in Python*. Springer.

Larose, C. D., & Larose, D. T. (2019). *Data Science using Python and R* (p. 176). Wiley.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830

References

Bruce, P., Bruce, A., & Gedeck, P. (2019). Practical Statistics for Data Scientists: 50+ Essential Concepts Using R and Python (2nd ed.). O'Reilly Media.

Chouinard, J. C. (2023, December 5). *What are PCA Loadings (with Python Example)*?JC Chouinard. Retrieved November 20, 2024, from <https://www.jcchouinard.com/pca-loadings/>

Firmin, S. (2019, July 29). *Tidying up with PCA: An Introduction to Principal Components Analysis*. Medium- Towards Data Science. Retrieved November 20, 2024, from <https://towardsdatascience.com/tidying-up-with-pca-an-introduction-to-principal-components-analysis-f876599af383?gi=40e97fbee3ee>

Gramfort, A., Blondel, M., & Grisel, O. (2022, October 18). StandardScaler. Scikit Learn. Retrieved November 15, 2024, from <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html>

James, G., Witten, D., Hastie, T., Tibsharani, R., & Taylor, J. (2023). *An Introduction to Statistical Learning with Applications in Python*. Springer.

Kumar, D. (2018, December 25). *Introduction to Data Preprocessing in Machine Learning*. Towards Data Science. Retrieved April 15, 2024, from <https://towardsdatascience.com/introduction-to-data-preprocessing-in-machine-learning-a9fa83a5dc9d>

Larose, C. D., & Larose, D. T. (2019). Data Science using Python and R (p. 176). Wiley.

Lee, E., PhD (2004, April 8). *Secrets of PCA: A Comprehensive Guide to Principal Component Analysis with Python and Colab*. Medium. Retrieved November 20, 2024, from https://drlee.io/secrets-of-pca-a-comprehensive-guide-to-principal-component-analysis-with-python-and-colab-6f7f3142e721

Lund Research Ltd. (2018). *Principal Components Analysis (PCA) using SPSS Statistics*. Laerd Statistics. Retrieved November 20, 2024, from <https://statistics.laerd.com/spss-tutorials/principal-components-analysis-pca-using-spss-statistics.php>

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830

Skoczylis, J. (2022). *Reducing Data*. Stats Made Easy. Retrieved November 20, 24, from <https://www.statsmadeasy.com/stats-concepts/7-reducing-data>

Soetewey, A. (2019, December 30). Variable types and examples. Stats and R. Retrieved August 26, 2023, from <https://statsandr.com/blog/variable-types-and-examples/>