Data Mining II

PA2 Dimensionality Reduction Methods

Jason Willis

College of Information Technology,

Western Governors University

Dr. Kesselly Kamara

June 25th, 2022

Table of Contents for Each Rubric

Hospital Readmission Problem

For our chain of hospitals to lower readmission concerns, we need to identify patients who have increased risk of rehospitalization within a month of their release. According to Schuller (2020), non-obese adults were 21% less likely to be readmitted than obese adults. A readmission study by Gert, et. al. (2002) showed a correlation between longer initial hospital stays and readmission. Within the provided dataset, I'm leveraging these studies to help create my hypothetical question and shape my approach in finding potential patient groups with a statistically significant chance for readmission outcomes.

After viewing the provided medical_clean.csv data set and accompanying data dictionary, there seems to be some patient groupings which are aligned with the research mentioned above. For instance, the following patient data fields: Initial patient admin days, Total Charges, and Initial Says (inpatient) both caught my attention and were underscored by the research mentioned above. While my initial feelings towards these variables might make them feel related, are they?

A1 – Proposal of Question

From information about previous patients who were readmitted, can we ascertain the minimum number of principal variables for our patients?

A2 – Defined Goal

The goal of our analysis is to logically investigate the provided data set and, by leveraging principal component analysis and other techniques, reduce the dimensionality of the provided dataset while keeping the model's accuracy high.

B1 – Explanation of PCA

According to Larose (2019) "PCA seeks to account for the correlation structure of a set of predictor variables, using a smaller set of uncorrelated linear combinations of these variables, called components." The approach aims to reduce the original number of predictor variables while accounting for most predictions in the original set. By reducing the predictor set, redundancies like multicollinearity are removed. Additionally, reductions in predictor values also helps to decrease overfitting.

B2 – PCA Assumption:

As stated in the Principal Component Analysis (PCA) explanation above, the technique assumes reduced data set *k* can provide almost indistinguishable predictive results as compared to the original complete data set *m*.

C1 – Continuous Dataset Variables

The data set accurately identifies continuous (and discrete) dataset variables needed to answer the PCA question from part A1 (Figure 1, Figure 2).

Create DataFrame w/Number DataTypes Only

```
1 df_num = df.select_dtypes(include='number')
2 df_num.head()
```

	CaseOrder	Zip	Lat	Lng	Population	Children	Age	Income
0	1	35621	34.34960	-86.72508	2951	1	53	86575.93
1	2	32446	30.84513	-85.22907	11303	3	51	46805.99
2	3	57110	43.54321	-96.63772	17125	3	53	14370.14
3	4	56072	43.89744	-93.51479	2162	0	78	39741.49
4	5	23181	37.59894	-76.88958	5287	1	22	1209.56

5 rows × 23 columns

• • •

• • •

Create DataFrame w/Categorical DataTypes Only

```
1 df_cat = df.select_dtypes(exclude='number')
2 df_cat.head()
```

	Customer_id	Interaction	UID	City	St
0	C412403	8cd49b13-f45a-4b47-a2bd-173ffa932c2f	3a83ddb66e2ae73798bdf1d705dc0932	Eva	
1	Z919181	d2450b70-0337-4406-bdbb-bc1037f1734c	176354c5eef714957d486009feabf195	Marianna	

Figure 1 - Identifying Continuous and Discrete Variables

```
[159]: 1 pca_df_binary_standardized.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 40 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Lat                                   10000 non-null  float64
 1   Lng                                   10000 non-null  float64
 2   Population                            10000 non-null  float64
 3   Children                              10000 non-null  float64
 4   Age                                   10000 non-null  float64
 5   Income                                10000 non-null  float64
 6   VitD_levels                           10000 non-null  float64
 7   Doc_visits                            10000 non-null  float64
 8   Full_meals_eaten                      10000 non-null  float64
 9   vitD_supp                             10000 non-null  float64
 10  Initial_days                          10000 non-null  float64
 11  TotalCharge                           10000 non-null  float64
 12  Additional_charges                    10000 non-null  float64
 13  Area__Suburban                        10000 non-null  float64
 14  Area__Urban                           10000 non-null  float64
 15  Marital__Married                      10000 non-null  float64
 16  Marital__Never Married                10000 non-null  float64
 17  Marital__Separated                    10000 non-null  float64
 18  Marital__Widowed                      10000 non-null  float64
 19  Gender__Male                          10000 non-null  float64
 20  Gender__Nonbinary                     10000 non-null  float64
 21  Soft_drink__Yes                       10000 non-null  float64
 22  Initial_admin__Emergency Admission    10000 non-null  float64
 23  Initial_admin__Observation Admission  10000 non-null  float64
 24  HighBlood__Yes                        10000 non-null  float64
 25  Stroke__Yes                           10000 non-null  float64
 26  Complication_risk__Low                10000 non-null  float64
 27  Complication_risk__Medium             10000 non-null  float64
 28  Overweight__Yes                       10000 non-null  float64
 29  Arthritis__Yes                        10000 non-null  float64
 30  Diabetes__Yes                         10000 non-null  float64
 31  Hyperlipidemia__Yes                   10000 non-null  float64
 32  BackPain__Yes                         10000 non-null  float64
 33  Anxiety__Yes                          10000 non-null  float64
 34  Allergic_rhinitis__Yes                10000 non-null  float64
 35  Reflux_esophagitis__Yes               10000 non-null  float64
 36  Asthma__Yes                           10000 non-null  float64
 37  Services__CT Scan                     10000 non-null  float64
 38  Services__Intravenous                 10000 non-null  float64
 39  Services__MRI                         10000 non-null  float64
dtypes: float64(40)
memory usage: 3.1 MB
```

Figure 2 - Data Types

C2 – Standardization of Dataset Variables

During the preprocessing steps, unnecessary features were removed dummy variables (Figure 3), were created.

Prep Dummies Data

```
[52]:
df_temp = df[['Age', 'Gender', 'ReAdmis', 'VitD_levels', 'Doc_visits', 'vitD_supp', 'Initial_admin', \
              'HighBlood', 'Stroke', 'Complication_risk', 'Overweight', 'Arthritis', 'Diabetes', 'Hyperlipidemia', \
              'BackPain', 'Anxiety', 'Allergic_rhinitis', 'Reflux_esophagitis', 'Asthma', 'Services', 'Initial_days', \
              'TotalCharge', 'Additional_charges']]

[56]:
df_dummies = pd.get_dummies(df_temp, drop_first=True)
df_dummies.head()
```

[56]:

	Age	VitD_levels	Doc_visits	vitD_supp	Initial_days	TotalCharge	Additional_charges	Gender_Male	Gender_Nonbinary	ReAdmis_Yes	...	Diabetes_Yes	Hyperlipidemia_Yes	Ba
0	53	19.141466	6	0	10.585770	3726.702860	17939.403420	1	0	0	...	1	0	
1	51	18.940352	4	1	15.129562	4193.190458	17612.998120	0	0	0	...	0	0	
2	53	18.057507	4	0	4.772177	2434.234222	17505.192460	0	0	0	...	1	0	
3	78	16.576858	4	0	1.714879	2127.830423	12993.437350	1	0	0	...	0	0	
4	22	17.439069	5	2	1.254807	2113.073274	3716.525786	0	0	0	...	0	1	

5 rows × 28 columns

Figure 3 - Create Dummy Data

Additionally, the dataset was standardized using sklearn.preproccessing StandardScaler() function (Figure 4) and then exported as std_df.csv.

Calculating Principle Components

```
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA

# Standardizing Continuous Dataset
scaler = StandardScaler()
std_df = scaler.fit_transform(df_dummies)

pca = PCA()
print(pca.fit_transform(std_df))
```

```
[[-1.22345671e+00  1.38470718e+00 -8.99594122e-01 ... -8.02858050e-02
   2.95935943e-02 -4.74054638e-10]
 [-1.07180686e+00  1.05141455e+00 -1.10734035e+00 ...  5.92369348e-02
   5.00476368e-02 -1.58767282e-09]
 [-1.94004422e+00  1.10048864e+00  5.73019886e-01 ... -2.23728403e-01
   2.00594989e-02  3.79674304e-09]
 ...
 [ 2.26175452e+00  4.45697534e-01  4.88310313e-01 ... -8.21186227e-02
   9.77371257e-02  2.67002221e-08]
 [ 1.97986293e+00 -1.19281366e+00 -7.54375873e-01 ... -1.68552210e-01
  -9.31021652e-02  1.12305098e-07]
 [ 2.16046478e+00 -4.67578843e-01  1.54701307e+00 ... -5.68823370e-02
   7.47586277e-02  2.00799114e-07]]
```

Export Cleaned and Standardized Dataset

```
1 pd.DataFrame(std_df).to_csv('std_df.csv', index=False)
```

Figure 4 - Standardize and Export Continuous Dataset

Correlation Matrix of All Principle Components (Standardized)

```
1 sns.set(rc = {'figure.figsize':(25,15)})
2 sns.heatmap(pd.DataFrame(std_df).corr(), annot=True);
```

Figure 5 - Matrix of All Principle Components

D1 – Principal Components

The submission accurately determines the matrix of all of the principal components (Figure 5).

D2 – Identification of Total Number of Components

The submission accurately identifies the total number of principal components. (Figure 6). The ideal number of components are 3 with an accuracy of 95.5%.

n_components = 3

Optimal n_components Plot

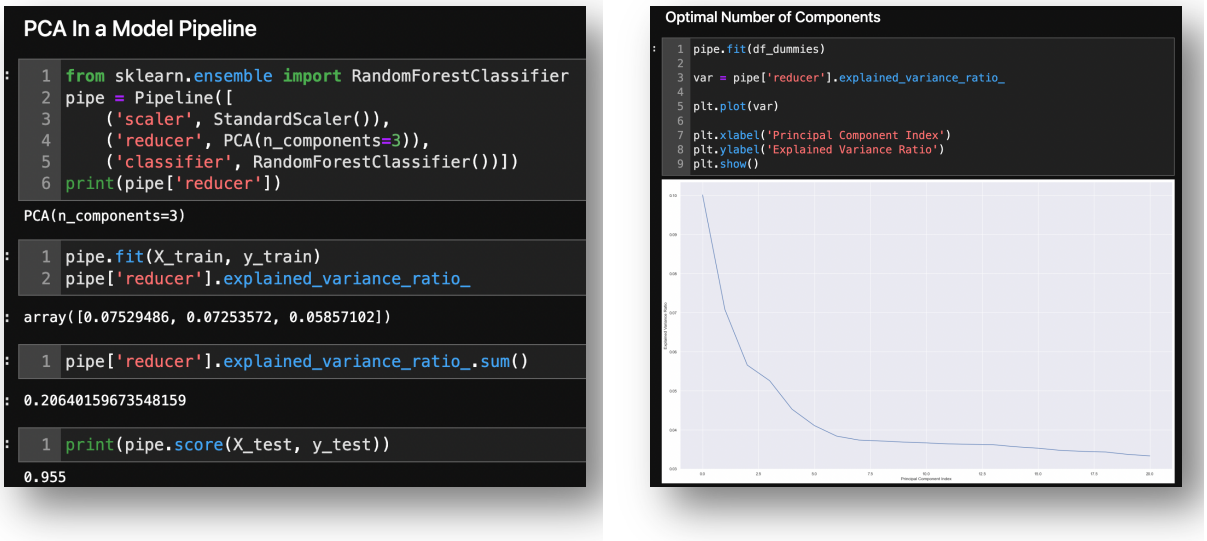

Figure 6 - Optimal n_components

D3 – Total Variance of Components

The submission accurately identifies the variance of each of the principal components identified in part D2 (Figure 6). You can see that each principal component's variance ratio and their sum.

D4 – Total Variance Captured by Components

The submission accurately identifies the total variance captured by the principal components identified in part D2 using the explained_variance_ratio_.sum(). (Figure 6)

D5 – Summary of Data Analysis

At 95.5% accuracy, this model is quite precise while reducing principal component features. I would have a high confidence in the dimension reduction process taken here to prepare the dataset. I believe we achieved our goal of figuring out the minimum number of principal variables for our patients while maintaining the model's accuracy high.

E – Sources for Third-Party Code

- Help using Markdown: https://www.markdownguide.org/basic-syntax/
- Help to see ALL columns: https://stackoverflow.com/questions/24524104/pandas-describe-is-not-returning-summary-of-all-columns
- Help to create a better histogram design: https://mode.com/example-gallery/python_histogram/
- Matplotlib Help: https://matplotlib.org/2.1.2/api/_as_gen/matplotlib.pyplot.plot.html
- Multiple ways to conduct ANOVA: https://www.marsja.se/four-ways-to-conduct-one-way-anovas-using-python/
- Numpy Help: https://numpy.org/doc/stable/
- Pandas Help: https://pandas.pydata.org/docs/user_guide/index.html#user-guide
- Python Help: https://docs.python.org/3.9/library/index.html
- Scipy.stats Help: https://docs.scipy.org/doc/scipy/reference/tutorial/stats.html

References

Gert P Westert, Ronald J Lagoe, Ilmo Keskimäki, Alastair Leyland, Mark Murphy, An international study of hospital readmissions and related utilization in Europe and the USA, Health Policy, Volume 61, Issue 3, 2002, Pages 269-278, ISSN 0168-8510, https://doi.org/10.1016/S0168-8510(01)00236-6. (https://www.sciencedirect.com/science/article/pii/S0168851001002366)

Larose, D., C., & Larose, D., T. (2019). Data Science Using Python and R. Wiley. https://www.wiley.com/en-us/Data+Science+Using+Python+and+R-p-9781119526810

Schuller K. A. (2020). Is obesity a risk factor for readmission after acute myocardial infarction? *Journal of healthcare quality research*, *35*(1), 4–11. https://doi.org/10.1016/j.jhqr.2019.09.002