Lora Milam
Western Governors University
D212 Data Mining II
10 March 2023

D212 Performance Assessment Task 2

1 Introduction

Lora Milam Masters Data Analytics (2/19/2023) Program Mentor:d212@wgu.edu

1.1 Research Question

This analysis will investigate leading factors within a readmission dataset for a popular medical hospital. The primary query is whether Principal Component Analysis(PCA) can assist in performing a dimensionality reduction when selecting primary features.

1.2 Research Goal

The goal of this analysis is to reduce the number of features associated with readmission.

2 Technique Justification

2.1 Explanation of PCA

In summary, PCA:

1. Creates a matrix containing direction and magnitude that determines the relationships between variables
2. Realigns data so that one of the components describes a majority of variance.
3. Remove variable with least variance

The expected outcome will be that the selected dataset will have less variables but still be able to retain the majority of the accuracy. Reduced variables will allow for reduced processing time and effort.

2.2 PCA Assumption

One assumption of the PCA is linearity, that all variables have a linear relationship(PCA).

3 Data Preparation

3.1 Dataset Variables

Variable	Type	Used in KMeans
Children	Continuous	Yes
Age	Continuous	Yes
Income	Continuous	Yes
VitD_levels	Continuous	Yes
Doc_visits	Continuous	Yes
Full_meals_eaten	Continuous	Yes
vitD_supp	Continuous	Yes
Initial_days	Continuous	Yes
TotalCharge	Continuous	Yes
Additional_charges	Continuous	Yes
ReAdmis	Categorical	No

In [1]:
```
# Libraries
import numpy as np
import pandas as pd
from pandas import Series, DataFrame
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
```

In [2]:
```
# Import dataset into Pandas dataframe
df = pd.read_csv('medical_clean_D212.csv')
df
```

Out[2]:

	CaseOrder	Customer_id	Interaction	UID	City	State	County	Zip	Lat	Lng	...	TotalCharge
0	1	C412403	8cd49b13-f45a-4b47-a2bd-173ffa932c2f	3a83ddb66e2ae73798bdf1d705dc0932	Eva	AL	Morgan	35621	34.34960	-86.72508	...	3726.702860
1	2	Z919181	d2450b70-0337-4406-bdbb-bc1037f1734c	176354c5eef714957d486009feabf195	Marianna	FL	Jackson	32446	30.84513	-85.22907	...	4193.190458
2	3	F995323	a2057123-abf5-4a2c-abad-8ffe33512562	e19a0fa00aeda885b8a436757e889bc9	Sioux Falls	SD	Minnehaha	57110	43.54321	-96.63772	...	2434.234222
3	4	A879973	1dec528d-eb34-4079-adce-0d7a40e82205	cd17d7b6d152cb6f23957346d11c3f07	New Richland	MN	Waseca	56072	43.89744	-93.51479	...	2127.830423
4	5	C544523	5885f56b-d6da-43a3-8760-83583af94266	d2f0425877b10ed6bb381f3e2579424a	West Point	VA	King William	23181	37.59894	-76.88958	...	2113.073274
...	...	...	...	...	...	...	...	...	...	...	...	...
9995	9996	B863060	a25b594d-0328-486f-a9b9-0567eb0f9723	39184dc28cc038871912ccc4500049e5	Norlina	NC	Warren	27563	36.42886	-78.23716	...	6850.942000
9996	9997	P712040	70711574-f7b1-4a17-b15f-48c54564b70f	3cd124ccd43147404292a883bf9ec55c	Milmay	NJ	Atlantic	8340	39.43609	-74.87302	...	7741.690000
9997	9998	R778890	1d79569d-8e0f-4180-a207-d67ee4527d26	41b770aeee97a5b9e7f69c906a8119d7	Southside	TN	Montgomery	37171	36.36655	-87.29988	...	8276.481000
9998	9999	E344109	f5a68e69-2a60-409b-a92f-ac0847b27db0	2bb491ef5b1beb1fed758cc6885c167a	Quinn	SD	Pennington	57775	44.10354	-102.01590	...	7644.483000
9999	10000	I569847	bc482c02-f8c9-4423-99de-3db5e62a18d5	95663a202338000abdf7e09311c2a8a1	Coraopolis	PA	Allegheny	15108	40.49998	-80.19959	...	7887.553000

10000 rows × 50 columns

In [3]:
```
# Review dataset
# Variables within dataset
df.columns
```

Out[3]:
```
Index(['CaseOrder', 'Customer_id', 'Interaction', 'UID', 'City', 'State',
       'County', 'Zip', 'Lat', 'Lng', 'Population', 'Area', 'TimeZone', 'Job',
       'Children', 'Age', 'Income', 'Marital', 'Gender', 'ReAdmis',
       'VitD_levels', 'Doc_visits', 'Full_meals_eaten', 'vitD_supp',
       'Soft_drink', 'Initial_admin', 'HighBlood', 'Stroke',
       'Complication_risk', 'Overweight', 'Arthritis', 'Diabetes',
       'Hyperlipidemia', 'BackPain', 'Anxiety', 'Allergic_rhinitis',
       'Reflux_esophagitis', 'Asthma', 'Services', 'Initial_days',
       'TotalCharge', 'Additional_charges', 'Item1', 'Item2', 'Item3', 'Item4',
       'Item5', 'Item6', 'Item7', 'Item8'],
      dtype='object')
```

In [4]:
```
# Summary stats of variables
df.describe()
```

Out[4]:

	CaseOrder	Zip	Lat	Lng	Population	Children	Age	Income	VitD_levels	Doc_visits	...	Ic
count	10000.00000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	...	100
mean	5000.50000	50159.323900	38.751099	-91.243080	9965.253800	2.097200	53.511700	40490.495160	17.964262	5.012200	...	53
std	2886.89568	27469.588208	5.403085	15.205998	14824.758614	2.163659	20.638538	28521.153293	2.017231	1.045734	...	21
min	1.00000	610.000000	17.967190	-174.209700	0.000000	0.000000	18.000000	154.080000	9.806483	1.000000	...	19
25%	2500.75000	27592.000000	35.255120	-97.352982	694.750000	0.000000	36.000000	19598.775000	16.626439	4.000000	...	31
50%	5000.50000	50207.000000	39.419355	-88.397230	2769.000000	1.000000	53.000000	33768.420000	17.951122	5.000000	...	52
75%	7500.25000	72411.750000	42.044175	-80.438050	13945.000000	3.000000	71.000000	54296.402500	19.347963	6.000000	...	74
max	10000.00000	99929.000000	70.560990	-65.290170	122814.000000	10.000000	89.000000	207249.100000	26.394449	9.000000	...	91

8 rows × 23 columns

In [5]:
```
# Determine if there are any missing values within dataset
df.isnull().sum()
```

Out[5]:
```
CaseOrder             0
Customer_id           0
Interaction           0
UID                   0
City                  0
State                 0
County                0
Zip                   0
Lat                   0
Lng                   0
Population            0
Area                  0
TimeZone              0
Job                   0
Children              0
Age                   0
Income                0
Marital               0
Gender                0
ReAdmis               0
VitD_levels           0
Doc_visits            0
Full_meals_eaten      0
vitD_supp             0
Soft_drink            0
Initial_admin         0
HighBlood             0
Stroke                0
Complication_risk     0
Overweight            0
Arthritis             0
Diabetes              0
Hyperlipidemia        0
BackPain              0
Anxiety               0
Allergic_rhinitis     0
Reflux_esophagitis    0
Asthma                0
Services              0
Initial_days          0
TotalCharge           0
Additional_charges    0
Item1                 0
Item2                 0
Item3                 0
Item4                 0
Item5                 0
Item6                 0
Item7                 0
Item8                 0
dtype: int64
```

In [6]:
```
# Review variable types
df.dtypes
```

Out[6]:
```
CaseOrder               int64
Customer_id            object
Interaction            object
UID                    object
City                   object
State                  object
County                 object
Zip                     int64
Lat                   float64
Lng                   float64
Population              int64
Area                   object
TimeZone               object
Job                    object
Children                int64
Age                     int64
Income                float64
Marital                object
Gender                 object
ReAdmis                object
VitD_levels           float64
Doc_visits              int64
Full_meals_eaten        int64
vitD_supp               int64
Soft_drink             object
Initial_admin          object
HighBlood              object
Stroke                 object
Complication_risk      object
Overweight             object
Arthritis              object
Diabetes               object
Hyperlipidemia         object
BackPain               object
Anxiety                object
Allergic_rhinitis      object
Reflux_esophagitis     object
Asthma                 object
Services               object
Initial_days          float64
TotalCharge           float64
Additional_charges    float64
Item1                   int64
Item2                   int64
Item3                   int64
Item4                   int64
Item5                   int64
Item6                   int64
Item7                   int64
Item8                   int64
dtype: object
```

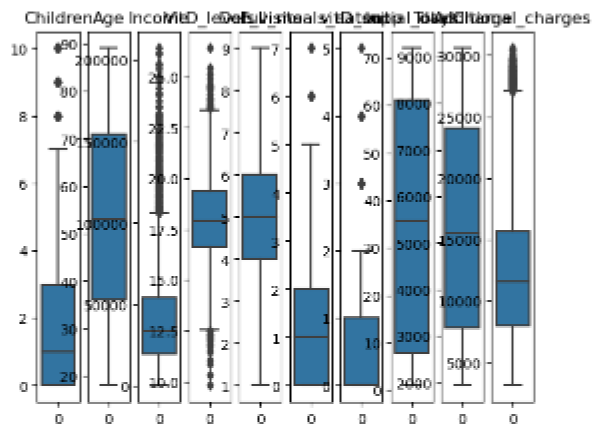
In [7]:
```
# Once you review the dataset
# Remove less meaningful and categorical variables
X=df.drop(columns=['CaseOrder', 'Customer_id', 'Interaction', 'UID', 'City', 'State',
       'County', 'Zip', 'Lat', 'Lng', 'Population', 'Area', 'TimeZone', 'Job', 'Marital', 'Gender',
       'Soft_drink', 'Initial_admin', 'HighBlood', 'Stroke', 'Complication_risk',
       'Overweight', 'Diabetes', 'Hyperlipidemia', 'BackPain',
       'Anxiety', 'Allergic_rhinitis', 'Reflux_esophagitis', 'Asthma',
       'Services', 'Arthritis', 'Item1', 'Item2', 'Item3', 'Item4', 'Item5',
       'Item6', 'Item7', 'Item8', 'ReAdmis'])
X.columns
```

Out[7]:
```
Index(['Children', 'Age', 'Income', 'VitD_levels', 'Doc_visits',
       'Full_meals_eaten', 'vitD_supp', 'Initial_days', 'TotalCharge',
       'Additional_charges'],
      dtype='object')
```

In [8]:

```
# Determine any outliers or discrepancies by reviewing univariate graphs
# Ouliers seem to be within reason
fig, axes = plt.subplots(ncols=len(X.columns))

# Create the boxplot with Seaborn
for column, axis in zip(X.columns, axes):
        sns.boxplot(data=X[column], ax=axis)
        axis.set_title(column)

# Show the plot
plt.tight_layout()
plt.show()
```

```
C:\Users\Mel\AppData\Local\Temp\ipykernel_72\3150235395.py:11: UserWarning: T
axes width small enough to accommodate all axes decorations
  plt.tight_layout()
```



In [9]:

```
# Convert the target variable (ReAdmis) to numeric and save it
y = pd.DataFrame()
y['ReAdmis'] = df['ReAdmis'].eq('Yes').mul(1)
y.head()
```

Out[9]:

	ReAdmis
0	0
1	0
2	0
3	0
4	0

3.2 Standardization of Dataset Variables

In [10]:

```
# Scale dataset
ss = StandardScaler()
ss.fit(X)
ss_data_array = ss.transform(X)
ss_data = pd.DataFrame(ss_data_array, columns = X.columns)
ss_data.head()
```

Out[10]:

	Children	Age	Income	VitD_levels	Doc_visits	Full_meals_eaten	vitD_supp	Initial_days	TotalCharge	Additional_charges
0	-0.507129	-0.024795	1.615914	0.583603	0.944647	-0.993387	-0.634713	-0.907310	-0.727185	0.765005
1	0.417277	-0.121706	0.221443	0.483901	-0.967981	0.990609	0.956445	-0.734595	-0.513228	0.715114
2	0.417277	-0.024795	-0.915870	0.046227	-0.967981	-0.001389	-0.634713	-1.128292	-1.319983	0.698635
3	-0.969332	1.186592	-0.026263	-0.687811	-0.967981	-0.001389	-0.634713	-1.244503	-1.460517	0.009004
4	-0.507129	-1.526914	-1.377325	-0.260388	-0.011687	-0.993387	2.547602	-1.261991	-1.467285	-1.408991

In [11]:

```
# Save prepared dataset for further analysis
ss_data.to_csv('D212_Part2_Scaled_Data.csv', index = False)
```

4 Analysis

4.1 Principal Components

In [12]:
```python
# Open scaled dataset
X_scaled = pd.read_csv("D212_Part2_Scaled_Data.csv")
```

In [13]:
```python
X_scaled.head()
```

Out[13]:

	Children	Age	Income	VitD levels	Doc visits	Full meals eaten	vitD supp	Initial days	TotalCharge	Additional charges
0	-0.507129	-0.024795	1.615914	0.583603	0.944647	-0.993387	-0.634713	-0.907310	-0.727185	0.765005
1	0.417277	-0.121706	0.221443	0.483901	-0.967981	0.990609	0.956445	-0.734595	-0.513228	0.715114
2	0.417277	-0.024795	-0.915870	0.046227	-0.967981	-0.001389	-0.634713	-1.128292	-1.319983	0.698635
3	-0.969332	1.186592	-0.026263	-0.687811	-0.967981	-0.001389	-0.634713	-1.244503	-1.460517	0.009004
4	-0.507129	-1.526914	-1.377325	-0.260388	-0.011687	-0.993387	2.547602	-1.261991	-1.467285	-1.408991

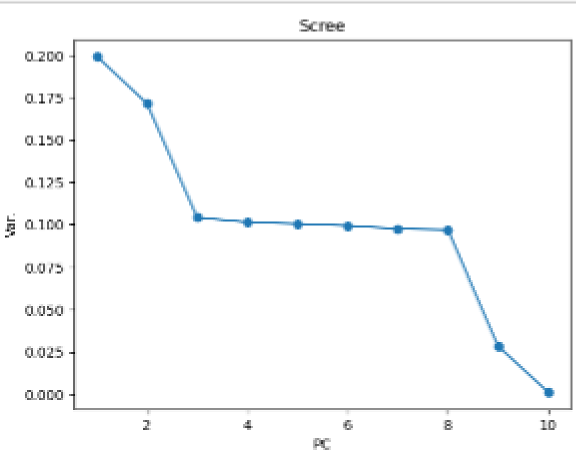
In [14]:
```python
# Identify PCA matrix
pca = PCA()
pc = pca.fit_transform(X_scaled)
var = pca.explained_variance_ratio_

print(pc)
```

```
[[-1.13597181  0.60909855 -0.79455115 ... -1.2282466  -0.57249721
   0.10093107]
 [-0.80308924  0.56810806  0.51559771 ... -0.15872543 -0.58763458
   0.12485647]
 [-1.63380069  0.68407439  0.75050855 ... -0.01999954 -0.50761057
  -0.16214324]
 ...
 [ 1.86757307 -0.28509127  0.54505622 ... -0.02158249 -0.53117442
  -0.02062733]
 [ 1.43209964 -1.0180605   0.95300084 ...  0.1560825   0.2068555
  -0.00909102]
 [ 1.97111758  0.21821058 -0.70470313 ...  0.03039824  0.71237735
  -0.11514686]]
```

4.2 Identification of Total Number of Components

The elbow plot shows that the optimal number of principal components is 9, since the variance between 9 and 10 is so low.

In [15]:
```python
# Identify PCA number using elbow rule
PC_values = np.arange(pca.n_components_) + 1
plt.plot(PC_values,var, 'o-')
plt.title('Scree')
plt.xlabel('PC')
plt.ylabel('Var.')
plt.show()
```



4.3 Total Variance of Components

```
In [16]: print(var)

[0.19943998 0.17146705 0.10412027 0.10137444 0.1004172  0.09934347
 0.09747879 0.09682792 0.0283594  0.00117147]

In [17]: print(var.cumsum())

[0.19943998 0.37090704 0.47502731 0.57640175 0.67681895 0.77616242
 0.87364121 0.97046913 0.99882853 1.        ]
```

4.4 Total Variance Captured by Components

The total variance captured with 9 features is 0.9988.

4.5 Summary of Data Analysis

In summary, PCA was able to reduce the number of features by one, while retaining a similar accuracy of the original 10. With that said, this reduction should allow for faster processing without increased resource spending.

5 Supporting Documentation

5.1 Video

This can be found within the attached file ‘Panopto Recording’.

5.2 Sources

A guide to principal component analysis (PCA) for Machine Learning. A Guide to Principal Component Analysis (PCA) for Machine Learning. (n.d.). Retrieved March 15, 2023, from https://www.keboola.com/blog/pca-machine-learning

Western Governors University. (n.d.). D212 Data Mining II. Salt Lake City.