D212 Dimensionality Reduction Methods Performance Assessment, Task 2

Alexa R. Fisher

Western Governors University

Degree: M.S. Data Analytics

Table of Contents

Part I: Research Question 3
A1. Proposal of Question 3
A2. Defined Goal 3
Part II: Method Justification 3
B1. Explanation of PCA 3
B2. PCA Assumption 4
Part III: Data Preparation 5
C1. Continuous Dataset Variables 5
C2. Standardization of Dataset Variables 6
Part IV: Analysis 8
D1. Principal Components 8
D2. Identification of Total Number of Components 10
D3. Total Variance of Components 11
D4. Total Variance Captured by Components 13
D5. Summary of Data Analysis 13
Part V: Attachments 14
E. Third-Party Web Sources 14
F. References 14

Part I: Research Question

A1. Proposal of Question

The research question for this thesis was, "Can the key attributes of the telecommunication customers be identified using PCA?". The analysis of the thesis was evaluated using the Principal Component Analysis or PCA. The variables evaluated were the continuous quantitative variables of "Population", "Children", "Age", "Income", "Outage_sec_perweek", "Email", "Contacts", "Yearly_equip_failure", "Tenure", "MonthlyCharge", and "Bandwidth_GB_Year". The purpose of PCA was to reduce the variables while still keeping as much information as possible (Whitfield, 2023). This could be useful in pinpointing key variables for the company to understand its customers. It would allow the telecommunication company to make better business decisions.

A2. Defined Goal

The goal of this analysis was to determine if PCA could pinpoint the key attributes of the telecommunication company's customers. Using principal component analysis, a dimensionality reduction method, allowed the telecommunication company to locate which variables were possibly redundant in the dataset. This would provide a narrowed scope of intel for the sales and marketing teams to construct viable strategies for new clientele and retention.

Part II: Method Justification

B1. Explanation of PCA

The Principal Component Analysis or PCA was described as a dimensionality reduction technique (Boeye, n.d.). This analysis allowed for the transformation of a dataset containing

several variables into a new set of variables. The new set of variables were linear combinations of the original called principal components (Whitfield, 2023). The principal components were selected in a way to capture the maximum amount of variation shown in the data. The main goal of PCA was to reduce the data via the reduction in its dimensionality while still keeping as much relevant data as possible. The technique identified and disregarded the least important principal components from the analysis. The remaining PCs would be kept and included in the final model.

The expected outcome of the principal component analysis would be a produced number of principal components. The number would be equal to the number of continuous or quantitative variables provided in the PCA. The value of each component would be concluded with the use of the "explained_variance_ratio_" within Python (Boeye, n.d.). The principal components deemed most important would be retained, while the least would be dropped. The reduction of dimensionality allowed for a decrease in the number of variables.

B2. PCA Assumption

There were various assumptions for the PCA. One of the main assumptions of PCA was linearity (Statistics Solutions, n.d.). The principal component analysis assumed there was a direct relationship between the variables within the dataset. The variables needed to show a linear correlation with each for principal component analysis to effectively represent the structure of the data within the loadings. Within the loadings, the relationships were identified be they positive or negative.

Part III: Data Preparation

C1. Continuous Dataset Variables

The PCA analysis used all of the quantitative variables within the telecommunication dataset provided by WGU. There was a total of eleven continuous variables. Please see the chart below to identify the variables as well as a brief description of each.

Variable	**Description**
Population	The census-based population is around a mile radius of the customer.
Children	The number of children a customer notated within their household at initial sign-up
Age	The age of the customer was reported at the initial sign-up.
Income	The total annual income reported by the customer at sign-up.
Outage_sec_perweek	The average weekly number of seconds of system outages within a customer's neighborhood.
Email	The count of emails sent to a client within the previous year.
Contacts	The total count of times a customer contacted technical support.

Yearly_equip_failure	The number of instances a customer's equipment failed and had to be repaired or replaced within the last year.
Tenure	The number of months a customer has been with the service provider.
MonthlyCharge	The monthly amount a customer was charged for service.
Bandwidth_GB_Year	The average amount of data used annually is noted in GB.

```
# define variables for the pca
df= data[['Population', 'Children', 'Age', 'Income', 'Outage_sec_perweek',
                'Email','Contacts', 'Yearly_equip_failure', 'Tenure','MonthlyCharge',
                'Bandwidth_GB_Year']]
```

```
df.head()
```

	Population	Children	Age	Income	Outage_sec_perweek	Email	Contacts	Yearly_equip_failure	Tenure	MonthlyCharge	Bandwidth_GB_Year
0	38.0	0.0	68	28561.99	7.978323	10.0	0.0	1.0	6.795513	172.455519	904.536110
1	10446.0	1.0	27	21704.77	11.699080	12.0	0.0	1.0	1.156681	242.632554	800.982766
2	3735.0	4.0	50	9609.57	10.752800	9.0	0.0	1.0	15.754144	159.947583	2054.706961
3	13863.0	1.0	48	18925.23	14.913540	15.0	2.0	0.0	17.087227	119.956840	2164.579412
4	11352.0	0.0	83	40074.19	8.147417	16.0	2.0	1.0	1.670972	149.948316	271.493436

C2. Standardization of Dataset Variables

Before performing the principal component analysis, the continuous variables need to be standardized. This was completed using the SciKit Learn Standard Scaler function. Standardization of the variables needed to be computed to make sure all the data was on the same scale. It was required to ensure the variables with larger ranges did not overpower those

with smaller ranges. This was completed with the use of the .StandardScaler() and .fit_transform() functions. The continuous variables utilized were “Population”, “Children”, “Age”, “Income”, “Outage_sec_perweek”, “Email”, “Contacts”, “Yearly_equip_failure”, “Tenure”, “MonthlyCharge”, “Bandwidth_GB_Year”. A function was created to loop through the variables to list the mean and standard deviation for each. This confirmed the data was accurately scaled across all variables. All the variables resulted in a mean of “0” and a standard deviation of “1.0”.

```
#scaling continuous variables.
scaler= StandardScaler()
df_std = scaler.fit_transform(df)
```

```
#function to notate mean/standard deviation of scaled df
for columns in col:
    col_avg = round(df_std_data.loc[:,columns].mean(), 2)
    col_std = round(df_std_data.loc[:,columns].std(), 2)
    print(f"{columns}: Mean = {col_avg} , Standard deviation = {col_std}.")
```

```
Population: Mean = 0.0 , Standard deviation = 1.0.
Children: Mean = 0.0 , Standard deviation = 1.0.
Age: Mean = -0.0 , Standard deviation = 1.0.
Income: Mean = -0.0 , Standard deviation = 1.0.
Outage_sec_perweek: Mean = 0.0 , Standard deviation = 1.0.
Email: Mean = -0.0 , Standard deviation = 1.0.
Contacts: Mean = 0.0 , Standard deviation = 1.0.
Yearly_equip_failure: Mean = 0.0 , Standard deviation = 1.0.
Tenure: Mean = 0.0 , Standard deviation = 1.0.
MonthlyCharge: Mean = -0.0 , Standard deviation = 1.0.
Bandwidth_GB_Year: Mean = 0.0 , Standard deviation = 1.0.
```

The cleaned scaled dataset was included: “AFCodeD212Tk2_clean.csv”.

```
#exporting cleaned scaled dataset
col=['Population', 'Children', 'Age', 'Income', 'Outage_sec_perweek',
                  'Email','Contacts', 'Yearly_equip_failure', 'Tenure','MonthlyCharge',
                  'Bandwidth_GB_Year']
df_std_data= pd.DataFrame(df_std, columns=col)
df_std_data.to_csv(r'AFCodeD212Tk2_clean.csv', index=False)
```

Part IV: Analysis

D1. Principal Components

After the standardization of the variables, the PCA could be performed. The principal component analysis was completed by using the PCA function to pass the number of components within it. The PCA computation was fitted and transformed on the standardized variables. The fit and transform were performed in a single step using the .fit_transform() function. A matrix of the principal components could be viewed below. It was visualized after printing the results stored in "loadings".

	PC1	PC2	PC3	PC4	PC5
Population	-0.013432	0.358192	0.218068	0.321068	0.414833
Children	0.014336	-0.482973	0.399498	-0.027158	0.139763
Age	0.001512	0.427676	-0.462298	0.046994	0.147135
Income	0.001715	-0.269166	-0.244079	-0.002601	-0.450979
Outage_sec_perweek	0.006114	0.201072	0.452415	-0.564588	0.102763
Email	-0.020448	0.216209	0.437521	0.428889	-0.013616
Contacts	0.003559	0.434301	-0.094654	-0.159681	-0.175570
Yearly_equip_failure	0.016245	-0.182210	-0.278065	-0.377251	0.664669
Tenure	0.705376	0.008896	-0.018055	0.046636	0.014998
MonthlyCharge	0.040509	0.272035	0.190872	-0.470188	-0.315791
Bandwidth_GB_Year	0.706881	-0.001611	0.020311	0.013729	-0.006715

	PC6	PC7	PC8	PC9	PC10
Population	0.213914	0.167080	0.653733	0.003175	0.218500
Children	0.155195	0.259128	-0.042098	0.699188	-0.065825
Age	-0.104545	0.419419	-0.100005	0.346159	-0.512194
Income	0.530524	0.515985	0.224337	-0.255402	-0.036452
Outage_sec_perweek	0.262424	-0.050689	0.059019	-0.248692	-0.537370
Email	0.067701	0.373466	-0.600077	-0.265631	0.050663
Contacts	0.598694	-0.270803	-0.257601	0.366564	0.337031
Yearly_equip_failure	0.129007	0.247625	-0.259652	-0.219428	0.334042
Tenure	0.018558	-0.022424	-0.006904	-0.019933	-0.030744
MonthlyCharge	-0.434624	0.430792	0.114566	0.103629	0.408273
Bandwidth_GB_Year	-0.000048	0.000375	0.002664	-0.002624	0.010693

	PC11
Population	-0.000143
Children	-0.021440
Age	0.022415
Income	-0.001029
Outage_sec_perweek	0.000315
Email	0.000226
Contacts	-0.000998
Yearly_equip_failure	-0.000040
Tenure	-0.705268
MonthlyCharge	-0.045728
Bandwidth_GB_Year	0.706783

The matrix indicated the variance weight or value of each attribute, which contributed to each of the eleven PCs. The featured attributes were referenced along the left x-axis. The PCs were lengthways across the top of the y-axis. The results of the matrix loadings provided insight into which attributes significantly influenced each principal component. For example, we can see "PC1" was strongly influenced by "Tenure" and "Bandwidth_GB_Year" as their weight was about 0.70. Strong significance was noted as values close to 1 or -1 (Boeye, n.d.).

The code to compute the PCA and create the matrix was shown below.

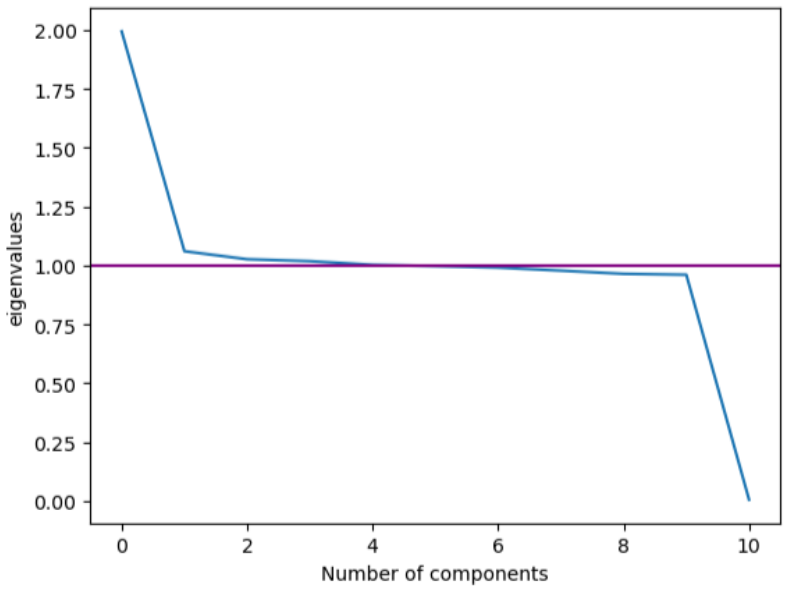
```
#pca
pca = PCA(n_components=df_std.shape[1],random_state=397)
pca_df=pca.fit_transform(df_std)
```

```
#pca loading chart
loadings = pd.DataFrame(pca.components_.T,
columns=['PC1', 'PC2', 'PC3', 'PC4', 'PC5', 'PC6', 'PC7',
         'PC8', 'PC9','PC10', 'PC11'],
index=col)
loadings
```

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11
Population	-0.013432	0.358192	0.218068	0.321068	0.414833	0.213914	0.167080	0.653733	0.003175	0.218500	-0.000143
Children	0.014336	-0.482973	0.399498	-0.027158	0.139763	0.155195	0.259128	-0.042098	0.699188	-0.065825	-0.021440
Age	0.001512	0.427676	-0.462298	0.046994	0.147135	-0.104545	0.419419	-0.100005	0.346159	-0.512194	0.022415
Income	0.001715	-0.269166	-0.244079	-0.002601	-0.450979	0.530524	0.515985	0.224337	-0.255402	-0.036452	-0.001029
Outage_sec_perweek	0.006114	0.201072	0.452415	-0.564588	0.102763	0.262424	-0.050689	0.059019	-0.248692	-0.537370	0.000315
Email	-0.020448	0.216209	0.437521	0.428889	-0.013616	0.067701	0.373466	-0.600077	-0.265631	0.050663	0.000226
Contacts	0.003559	0.434301	-0.094654	-0.159681	-0.175570	0.598694	-0.270803	-0.257601	0.366564	0.337031	-0.000998
Yearly_equip_failure	0.016245	-0.182210	-0.278065	-0.377251	0.664669	0.129007	0.247625	-0.259652	-0.219428	0.334042	-0.000040
Tenure	0.705376	0.008896	-0.018055	0.046636	0.014998	0.018558	-0.022424	-0.006904	-0.019933	-0.030744	-0.705268
MonthlyCharge	0.040509	0.272035	0.190872	-0.470188	-0.315791	-0.434624	0.430792	0.114566	0.103629	0.408273	-0.045728
Bandwidth_GB_Year	0.706881	-0.001611	0.020311	0.013729	-0.006715	-0.000048	0.000375	0.002664	-0.002624	0.010693	0.706783

D2. Identification of Total Number of Components

The identification of the total number of components was determined using the Kaiser Rule. The Kaiser rule noted only components with an eigenvalue of greater than or equal to "1" should be retained. The rule suggested that components with eigenvalues of under "1" would explain less variance (Unknown, 2017). A scree plot of the eigenvalues was shown below. A line was marked across the x-axis at "1" to visually see which components fell below the noted threshold.



The visualization showed components after “PC6” were below the criterion specified by the Kaiser rule. The application of the Kaiser rule suggested “PC7” through “PC11” should be dropped. The total number of components used for the final loadings would be six.

D3. Total Variance of Components

After the determination that six principal components were ideal for the analysis, the same analysis steps were taken to show the final data. The PCA procedure was completed reflecting the number of components as “6”. It was fitted and transformed using .fit_transform(). The PCA final components were placed into a dataframe and loaded to show a chart of the new loadings.

```
#reduced PC
pca_final = PCA(n_components=6, random_state=397)
pca_final.fit_transform(df_std)
```

```
array([[-1.52627426, -0.28249885, -1.60166668, -0.3681253 ,  0.58805934,
        -1.33897586],
       [-1.65934365, -0.1086178 ,  0.9681424 , -1.36345462,  0.49565388,
        -1.32572172],
       [-0.89304657, -1.25776437,  0.08533259, -0.86171821,  1.41318147,
        -0.89928314],
       ...,
       [ 0.58494326, -0.90990097, -0.88054697,  0.58201517, -0.87859325,
        -1.07155638],
       [ 1.9869556 ,  1.90444291,  1.94191984,  0.02746571,  0.19446855,
        -0.6357024 ],
       [ 1.55203045,  1.06572196,  2.08910562,  0.26601863, -0.28318707,
        -0.74396204]])
```

```
#final loadings
loadings = pd.DataFrame(pca_final.components_.T,
columns=['PC1', 'PC2', 'PC3', 'PC4', 'PC5', 'PC6'],
index=df.columns)
loadings
```

	PC1	PC2	PC3	PC4	PC5	PC6
Population	-0.013432	0.358192	0.218068	0.321068	0.414833	0.213914
Children	0.014336	-0.482973	0.399498	-0.027158	0.139763	0.155195
Age	0.001512	0.427676	-0.462298	0.046994	0.147135	-0.104545
Income	0.001715	-0.269166	-0.244079	-0.002601	-0.450979	0.530524
Outage_sec_perweek	0.006114	0.201072	0.452415	-0.564588	0.102763	0.262424
Email	-0.020448	0.216209	0.437521	0.428889	-0.013616	0.067701
Contacts	0.003559	0.434301	-0.094654	-0.159681	-0.175570	0.598694
Yearly_equip_failure	0.016245	-0.182210	-0.278065	-0.377251	0.664669	0.129007
Tenure	0.705376	0.008896	-0.018055	0.046636	0.014998	0.018558
MonthlyCharge	0.040509	0.272035	0.190872	-0.470188	-0.315791	-0.434624
Bandwidth_GB_Year	0.706881	-0.001611	0.020311	0.013729	-0.006715	-0.000048

Lastly, the final data was run through the same “explained_variance_ratio_” method to provide a chart of the variance contributions for this reduced set. These values were the same as the previous chart excluding the removed “PC7” through “PC11” variance contribution percentages.

```
#chart of captured variance per final PC
varx_final= pca_final.explained_variance_ratio_ * 100
varx_final_df = pd.DataFrame(varx_final.round(2), columns=['Variance Percentage Per PC, Final'],
                        index=['PC1', 'PC2', 'PC3', 'PC4', 'PC5', 'PC6'])
varx_final_df
```

	Variance Percentage Per PC, Final
PC1	18.13
PC2	9.64
PC3	9.33
PC4	9.26
PC5	9.12
PC6	9.07

D4. Total Variance Captured by Components

The total variance captured by the principal components was identified for the final data. This was shown to be ~64.55%. It was consistent with the previous data notated. The initial eleven principal components accounted for 100% of the variance in the dataset. The total percentage from "PC7" through "PC11" variance contribution was calculated to be 35.45%. The reduction of the principal components matched the calculation of the total 100% variance minus the 35.45% from the removed components. (100% – 35.45% = 64.55%)

```
#printing total percentage of final pc explained variances
print(sum(pca_final.explained_variance_ratio_ * 100).round(3))
```

```
64.545
```

D5. Summary of Data Analysis

The results of the data analysis could be summarized in the following point: PCA can identify key attributes. The usage of PCA could be beneficial in the reduction of the multi-dimensionality of data. The key attributes of the telecommunication customers could be identified using this technique. It provided a helpful visualization of the dimensions as well as compressed the information into components to de-clutter the data. In the case of this telecommunication dataset, there were six principal components. The first principal component had a positive relationship between "Tenure" and "Bandwidth_GB_Year" with "0.7053" and "0.7068" coefficients. The second component had a positive relationship with "Population", "Age", and "Contacts" as noted based on their coefficients of "0.3581", "0.4276", and "0.4343". The opposite was true in comparison for "Children" where it had a negative relationship based on its coefficient of "-0.4829". The third component had a positive relationship with "Children", "Outage_sec_perweek", and "Email", but a negative relationship with "Age". These are all

examples of the results that could be ascertained by the analysis of the final PCA loadings. The relationship positivity or negativity directly correlated to if the attribute increased or decreased with each additional customer. Knowing this information would be beneficial in identifying key attributes of the company's customers.

Part V: Attachments

E. Third-Party Web Sources

Kamara, Kesselly.(n.d.). PCA Webinar Recording [Video] WGU Hosted Panopto: https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=60fa4159-94ba-4f41-9ba8-aea0011f18f9

Unknown.(2021, March 15). Python Pandas -highlighting cells in a dataframe. StackOverflow: Python Pandas - highlighting cells in a dataframe - Stack Overflow

F. References

Boeye, J. (n.d.). *Dimensionality Reduction in Python*. Retrieved from DataCamp: https://app.datacamp.com/learn/courses/dimensionality-reduction-in-python

Statistics Solutions. (n.d.). *Principal Component Analysis (PCA)*. Retrieved from Statistics Solutions: https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/principal-component-analysis-pca/

Unknown. (2017, February 28). *Kaiser Rule.* Retrieved from displayr: https://docs.displayr.com/wiki/Kaiser_Rule

Whitfield, B. (2023, March 29). *Principal Component Analysis*. Retrieved from Built In: https://builtin.com/data-science/step-step-explanation-principal-component-analysis