Dimension Reduction of Telcom Customer Churn Data

Mike Mattinson

Western Governors University

D212: Data Mining II

Task 2: Dimension Reduction

Dr. Kesselly Kamara

May 13, 2022

Revision 6

Abstract

Telecom customer data will be analyzed for feature selection using principal component analysis (PCA). The dataset consists of 50 features associated with 10,000 customer records. The purpose of this analysis is reduce number of features by finding and removing irrelevant data. The analysis will reduce the number of principal components to 39. While using Kaiser criterion, the analysis will reduce the number of principal components to 26.

Keywords: Telecom. Churn. Data Mining. Dimension Reduction. PCA.

Scenario 1

One of the most critical factors in customer relationship management that directly affects a company’s long-term profitability is understanding its customers. When a company can better understand its customer characteristics, it is better able to target products and marketing campaigns for customers, resulting in better profits for the company in the long term.

You are an analyst for a telecommunications company that wants to better understand the characteristics of its customers. You have been asked to use principal component analysis (PCA) to analyze customer data to identify the principal variables of your customers, ultimately allowing better business and strategic decision-making.

List of Tables

[Table 1 *First 5 rows of the original dataset* 8](#_Toc103374192)

[Table 2 *First 5 rows of the scaled data* 10](#_Toc103374193)

[Table 3 *Load factors using first four principal components* 18](#_Toc103374194)

List of Figures

[Figure 1 *Define continuous features* 7](#_Toc103374174)

[Figure 2 *D1. Determine matrix of all the principal components* 12](#_Toc103374175)

[Figure 3 *D2. Use scree plot to identify variance for each component* 14](#_Toc103374176)

[Figure 4 *D4. Use cumulative variance to identify total variance as function of components* 16](#_Toc103374177)

# Part I: Research Question

A. Describe the purpose of this data mining report by doing the following:

## A1. Propose one question relevant to a real-world organizational situation that you will answer using principal component analysis (PCA)

What are the most influential features of the Telecom customer data related to churn?

## A2. Define one goal of the data analysis. Ensure that your goal is reasonable within the scope of the scenario and is represented in the available data.

Use principal component analysis (PCA) to reduce dimensionality of the problem set to a more manageable number of principal components. The primary dataset consists of 10,000 customer records with 50 attributes each, which by definition is a high dimension dataset. The overall steps to the analysis are below:

# Part II: Method Justification

B. Explain the reasons for using PCA by doing the following:

## B1. Explain how the PCA analyzes the selected the selected dataset. Include expected outcomes.

According to Vadapalli (2020), “Principal component analysis (PCA) is a statistical method used to transform a large number of possibly correlated variables into a much smaller number of uncorrelated variables referred to as principal components. PCA can be used as a data reduction technique as it allows us to find the most important variables that are needed to describe a dataset. PCA can also be used to reduce the dimensionality of the data space in order to get insight on the inner structure of the data. This is helpful when dealing with large datasets.”

The expected outcome of the reduction analysis is the appropriate number of principal components and total variance for each component.

## B2. Summarize one assumption of PCA.

PCA is dependent on having numeric, scaled data. If any one feature is not scaled and has large values, the PCA will give more weight to those higher values. In order to place equal weight of all features, all of the data should be scaled such that the mean of the feature is 0 and the standard deviation is 1.

# Part III: Data Preparation

C. Perform data preparation for the chosen dataset by doing the following:

## C1. Identify the continuous dataset variables that you will need in order to answer the PCA question proposed in part A1.

The list of continuous features that will be used for the PCA is shown in Figure 1 and Figure 2. They are:

* Children
* Population
* Age
* Income
* Tenure
* Email
* Contacts
* Outage\_sec\_perweek
* MonthlyCharge
* Bandwidth\_GB\_Year
* Lat
* Lng

All remaining data is either categorical or discreet and will not be used for the PCA.

Figure   
Define continuous features

Graphical user interface, text, application, email

Description automatically generated

Notes.

Table 1  
First 5 rows of the original dataset

0 1 2 3 4

Children 0.000 1.000 4.000 1.000 0.000

Population 38.000 10446.000 3735.000 13863.000 11352.000

Age 68.000 27.000 50.000 48.000 83.000

Income 28561.990 21704.770 9609.570 18925.230 40074.190

Tenure 6.796 1.157 15.754 17.087 1.671

Email 10.000 12.000 9.000 15.000 16.000

Contacts 0.000 0.000 0.000 2.000 2.000

Outage\_sec\_perweek 7.978 11.699 10.753 14.914 8.147

MonthlyCharge 172.456 242.633 159.948 119.957 149.948

Bandwidth\_GB\_Year 904.536 800.983 2054.707 2164.579 271.493

Lat 56.251 44.329 45.356 32.967 29.380

Lng -133.376 -84.241 -123.247 -117.248 -95.807

Notes. Info(): <class 'pandas.core.frame.DataFrame'>

RangeIndex: 10000 entries, 0 to 9999

Data columns (total 12 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Children 10000 non-null int64

1 Population 10000 non-null int64

2 Age 10000 non-null int64

3 Income 10000 non-null float64

4 Tenure 10000 non-null float64

5 Email 10000 non-null int64

6 Contacts 10000 non-null int64

7 Outage\_sec\_perweek 10000 non-null float64

8 MonthlyCharge 10000 non-null float64

9 Bandwidth\_GB\_Year 10000 non-null float64

10 Lat 10000 non-null float64

11 Lng 10000 non-null float64

dtypes: float64(7), int64(5)

memory usage: 937.6 KB

## C2. Standardize the continuous dataset variables identified in part C1. Include a copy of the cleaned dataset.

Scaled data is critical to the PCA. If not scaled properly, larger values will tend to dominate the PCA. Table 2 shows the first five rows of the scaled data. The StandardScaler was used to fit and transform the raw data.

Table   
First 5 rows of the scaled data

|  |
| --- |
| 0 1 2 3 4  Children -0.972 -0.507 0.891 -0.507 -0.972  Population -0.673 0.048 -0.417 0.285 0.111  Age 0.721 -1.260 -0.149 -0.245 1.446  Income -0.399 -0.642 -1.071 -0.741 0.009  Tenure -1.049 -1.262 -0.710 -0.660 -1.243  Email -0.666 -0.005 -0.997 0.986 1.317  Contacts -1.006 -1.006 -1.006 1.018 1.018  Outage\_sec\_perweek -0.680 0.570 0.252 1.651 -0.623  MonthlyCharge -0.004 1.630 -0.295 -1.227 -0.528  Bandwidth\_GB\_Year -1.138 -1.186 -0.612 -0.562 -1.428  Lat 3.217 1.025 1.214 -1.065 -1.725  Lng -2.810 0.432 -2.142 -1.746 -0.332  <class 'pandas.core.frame.DataFrame'>  RangeIndex: 10000 entries, 0 to 9999  Data columns (total 12 columns):  # Column Non-Null Count Dtype  --- ------ -------------- -----  0 Children 10000 non-null float64  1 Population 10000 non-null float64  2 Age 10000 non-null float64  3 Income 10000 non-null float64  4 Tenure 10000 non-null float64  5 Email 10000 non-null float64  6 Contacts 10000 non-null float64  7 Outage\_sec\_perweek 10000 non-null float64  8 MonthlyCharge 10000 non-null float64  9 Bandwidth\_GB\_Year 10000 non-null float64  10 Lat 10000 non-null float64  11 Lng 10000 non-null float64  dtypes: float64(12)  memory usage: 937.6 KB  None |

Notes. The data is available as an attached file located at tables\scaled\_df.csv

# Part IV: Analysis

D. Perform the data analysis and report on the results by doing the following:

1. Determine the matrix of all the principal components. (Figure 1). Generate a correlation matrix of all numerical data, and look for correlation values greater than 0.70. Evaluate the highly correlated features and consider removing one or the other. (Bex, 2021)
2. Identify the total number of principal components using the elbow rule or the Kaiser criterion. Include a screenshot of the scree plot. (Figure 2). Use the PCA components, eigenvalues and variance values to generate a scree plot. Evaluate where on the plot represents the optimum number of components. Consider the Kaiser criterion which says to drop features where the eigenvalue is less than 1.
3. Identify the variance of each of the principal components identified in part D2. (Figure 3). Using the PCA components and variance values, generate a plot showing amount of variance for each principal component.
4. Identify the total variance captured by the principal components identified in part D2. (Figure 4). Using the PCA component variance, generate a cumulative plot of total variance. Find the minimum number of principal components that exceed a total variance of 85%, 90% or 95%.
5. Summarize the results of your data analysis.

Figure 2  
D1. Determine matrix of all the principal components

A picture containing chart

Description automatically generated

Notes. Figure 1 is the correlation matrix for all numerical data, including the newly created categorical dummy features. To help identify the features with high correlation, the names of the features with correlation above 0.7 are listed in the figure notes.

['Tenure', 'Gender\_Female']

Here is the adapted code to generate the correlation matrix in Figure 1 (Bex, 2021):

# Create a mask

# adapted code (Bex, 2021)

matrix **=** D**.**corr**()**

plt**.**figure**(**figsize**=(**16**,**12**))**

cmap **=** sns**.**diverging\_palette**(**250**,** 15**,** s**=**75**,** l**=**40**,**

n**=**9**,** center**=**"light"**,** as\_cmap**=True)**

mask **=** np**.**triu**(**np**.**ones\_like**(**matrix**,** dtype**=bool))**

sns**.**heatmap**(**matrix**,** mask**=**mask**,** center**=**0**,** annot**=True,**

fmt**=**'.2f'**,** square**=True,** cmap**=**cmap**)**

Figure 3  
D2. Use scree plot to identify variance for each component

Chart, line chart

Description automatically generated

(10000, 52)

(10000, 39)

Notes. Figure 2 shows the scree plot. PCA was able to successfully reduce the problem dimension from 52 to 39. Although the first two components have the most explained variance, it takes all of the 39 components to explain 95% (see Figure 3).

Here is the code to create the scree plot:

# PCA - keep 90% of variance (Boyle, 2019)

pca **=** PCA**(**0.95**)**

principal\_components **=** pca**.**fit\_transform**(**X**)**

principal\_df **=** pd**.**DataFrame**(**data **=** principal\_components**)**

**print(**A**.**shape**)**

**print(**principal\_df**.**shape**)**

## D3. Identify the variance of each of the principal components identified in part D2.

The explained variance ratios for each principal component:

array([0.43319022, 0.06947163, 0.02696186, 0.02574512, 0.02300794,

0.02293276, 0.02062869, 0.01795037, 0.01764734, 0.01757722,

0.01734662, 0.01708094, 0.01696258, 0.01688812, 0.01679839,

0.01648529, 0.0163014 , 0.01589637, 0.01569515, 0.01559179,

0.0152735 , 0.01520301, 0.0146579 , 0.01424522, 0.01403923,

0.01363732, 0.01341701, 0.01315427])

Code used to calculated these ratios:

pca**.**explained\_variance\_ratio\_

The explained variance for each principal component also known as the eigenvalues:

array([**2.95331466, 2.09708838, 1.72647177, 1.6464298 , 1.6158704 ,**

**1.51286066, 1.5089432 , 1.4396131 , 1.35090479, 1.31073554,**

**1.29621584, 1.28024198, 1.25996635, 1.25322462, 1.24445112,**

**1.22687776, 1.15499185, 1.06679257, 1.05330037, 1.04799724,**

**1.03641008, 1.03441197, 1.02859453, 1.022224 , 1.01408408,**

**1.00405837**, 0.99857262, 0.98238462, 0.97787214, 0.97041187,

0.96866228, 0.95769867, 0.95013982, 0.93132333, 0.92690545,

0.90929756, 0.84158254, 0.77122477, 0.68610635])

Notes. The Kaiser criterion suggests that you select eigenvalues above 1, which are the ones highlighted, so it would suggest using 26 principal components.

Code used to calculated these eigenvalues:

pca**.**explained\_variance\_

Figure 4  
D4. Use cumulative variance to identify total variance as function of components

Chart, line chart

Description automatically generated

Notes.

Code used to generate Figure 3 (Tripathi, 2019):

# create cumulative variance plot (Tripathi, 2019)

**%**matplotlib notebook

plt**.**figure**(**figsize **=** **(**8**,**4**))**

plt**.**plot**(**np**.**cumsum**(**pca**.**explained\_variance\_ratio\_**\***100**))**

plt**.**xlim**(**xmax **=** 50**,** xmin **=** 0**)**

plt**.**ylim**(**ymax **=** 100**,** ymin **=** 0**)**

plt**.**title**(**'Cumulative Variance'**)**

plt**.**xlabel**(**'number of components'**)**

plt**.**ylabel**(**'cumulative explained variance'**);**

plt**.**axhline**(**y **=** 95**,** color**=**'k'**,** linestyle**=**'--'**,** label **=** '95% Variance'**)**

plt**.**axhline**(**y **=** 90**,** color**=**'c'**,** linestyle**=**'--'**,** label **=** '90% Variance'**)**

plt**.**axhline**(**y **=** 85**,** color**=**'r'**,** linestyle**=**'--'**,** label **=** '85% Variance'**)**

plt**.**legend**(**loc**=**'best'**)**

plt**.**show**()**

## D5. Summarize the results of your data analysis.

Figure 3 shows a total variance of 95% using 39 principal components, a nice reduction in overall dimensionality from the original 50 attributes. However, using the Kaiser criterion, the reduction should be 26 principal components with an overall total variance of about 70%.

In addition, Table 2 below shows the load factors of each of the first four principal components for consideration. Table 2 is now available to analyze for further considerations.

Table 3  
Load factors using first four principal components

|  | **PC-1** | **PC-2** | **PC-3** | **PC-4** |
| --- | --- | --- | --- | --- |
| **Children** | 0.004287 | -0.011847 | 0.056988 | -0.007169 |
| **Age** | 0.006379 | 0.010735 | -0.040605 | -0.002853 |
| **Income** | 0.001256 | -0.012560 | 0.006080 | 0.015310 |
| **Outage\_sec\_perweek** | -0.017785 | 0.028091 | 0.012807 | -0.009239 |
| **Email** | 0.008474 | 0.003132 | -0.011032 | -0.007576 |
| **Contacts** | -0.008854 | 0.010380 | -0.003544 | -0.008951 |
| **Yearly\_equip\_failure** | -0.007175 | -0.010055 | 0.016260 | -0.002627 |
| **MonthlyCharge** | 0.004897 | 0.651798 | 0.017227 | 0.006228 |
| **Bandwidth\_GB\_Year** | -0.008765 | 0.042386 | 0.067694 | 0.007396 |
| **Item1** | 0.458462 | 0.002105 | 0.065232 | 0.266118 |
| **Item2** | 0.433380 | -0.006572 | 0.063387 | 0.271625 |
| **Item3** | 0.400054 | -0.017077 | 0.075854 | 0.261363 |
| **Item4** | 0.145856 | 0.000282 | -0.138714 | -0.538065 |
| **Item5** | -0.175411 | -0.008643 | 0.151248 | 0.553502 |
| **Item6** | 0.404361 | -0.000385 | -0.041612 | -0.174481 |
| **Item7** | 0.357802 | -0.010872 | -0.038490 | -0.173017 |
| **Item8** | 0.307872 | -0.003411 | -0.047196 | -0.110907 |
| **Techie** | 0.007654 | 0.005402 | 0.012696 | 0.025636 |
| **Port\_modem** | 0.001061 | 0.003508 | -0.018632 | -0.010168 |
| **Tablet** | 0.016664 | 0.005912 | 0.032047 | -0.005458 |
| **Phone** | 0.005372 | -0.024208 | 0.025749 | 0.020751 |
| **Multiple** | 0.000647 | 0.239813 | 0.047058 | 0.002902 |
| **OnlineSecurity** | 0.001275 | 0.046507 | 0.024437 | -0.020494 |
| **OnlineBackup** | -0.004294 | 0.155624 | 0.004595 | 0.038125 |
| **DeviceProtection** | -0.003015 | 0.112385 | 0.005802 | 0.016252 |
| **TechSupport** | 0.024767 | 0.049147 | 0.006000 | -0.011838 |
| **StreamingTV** | 0.000668 | 0.284363 | 0.047628 | -0.019655 |
| **StreamingMovies** | -0.005987 | 0.365867 | -0.002556 | -0.013579 |
| **PaperlessBilling** | 0.005735 | 0.006693 | 0.007231 | -0.014428 |
| **Area\_Rural** | 0.014564 | -0.001592 | 0.019460 | -0.011710 |
| **Area\_Suburban** | -0.002729 | -0.005928 | 0.012699 | -0.056373 |
| **Area\_Urban** | -0.011831 | 0.007529 | -0.032177 | 0.068163 |
| **Marital\_Divorced** | -0.005619 | -0.003829 | -0.026831 | -0.060881 |
| **Marital\_Married** | -0.002297 | -0.013400 | -0.010474 | 0.021426 |
| **Marital\_Never Married** | 0.014242 | 0.010585 | -0.015655 | -0.065502 |
| **Marital\_Separated** | 0.008360 | 0.020935 | 0.014914 | -0.005697 |
| **Marital\_Widowed** | -0.014461 | -0.014349 | 0.037958 | 0.110955 |
| **Gender\_Male** | -0.011993 | 0.010563 | -0.002120 | 0.004230 |
| **Gender\_Nonbinary** | 0.007434 | -0.000591 | 0.016339 | -0.011388 |
| **Contract\_Month-to-month** | -0.005073 | -0.032387 | -0.715505 | 0.189692 |
| **Contract\_One year** | 0.005451 | 0.018520 | 0.370886 | -0.082303 |
| **Contract\_Two Year** | 0.000710 | 0.019972 | 0.477510 | -0.141793 |
| **InternetService\_DSL** | -0.004315 | -0.194057 | 0.139568 | -0.016785 |
| **InternetService\_Fiber Optic** | 0.016730 | 0.390904 | -0.127339 | 0.038016 |
| **InternetService\_None** | -0.015276 | -0.248552 | -0.007776 | -0.026597 |
| **PaymentMethod\_Bank Transfer(automatic)** | -0.004483 | 0.007619 | 0.023450 | -0.015990 |
| **PaymentMethod\_Credit Card (automatic)** | -0.002710 | 0.031776 | -0.026254 | -0.048399 |
| **PaymentMethod\_Electronic Check** | 0.010785 | -0.026531 | 0.026856 | 0.112633 |
| **PaymentMethod\_Mailed Check** | -0.005098 | -0.008351 | -0.028126 | -0.064348 |

Notes. For example, the first principal component (PC-1) has its strongest loadings from the following six features:

* Item1
* Item2
* Item3
* Item 6
* Item 7
* Item 8

The sign of the loading value doesn’t matter, but just the magnitude (abs(x)). For PC-2, the strongest loadings come from the following six features:

* MonthlyCharge
* Multiple
* OnlineSecurity
* StreamingMovies
* Internet\_Fiber
* Internet\_None

# Part V: Attachments

E. Record the web sources used to acquire data or segments of third-party code to support the analysis. Ensure the web sources are reliable. (see References below)

F. Acknowledge sources, using in-text citations and references, for content that is quoted, paraphrased, or summarized. (see References below)

G. Demonstrate professional communication in the content and presentation of your submission.

References

Albon, C. (2017). Drop Highly Correlated Features. Retrieved from https://chrisalbon.com/code/machine\_learning/feature\_selection/drop\_highly\_correlated\_features/

Alkhayrat, M., Aljnidi, M., & Aljoumaa, K. (2020). A comparative dimensionality reduction study in telecom customer segmentation using deep learning and PCA. Retrieved from https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-0286-0

Arvai, K. (2022). K-Means Clustering in Python: A Practical Guide. Retrieved from https://realpython.com/k-means-clustering-python/#: :text=The SSE is defined as,try to minimize this value.&text=The purpose of this figure,centroids is an important step.

Bex, T. (2021, April 13). *How to Use Pairwise Correlation For Robust Feature Selection*. Retrieved May 8, 2022, from How to Use Pairwise Correlation For Robust Feature Selection: https://towardsdatascience.com/how-to-use-pairwise-correlation-for-robust-feature-selection-20a60ef7d10

Boyle, T. (2019, March 25). *Feature Selection and Dimensionality Reduction*. Retrieved May 8, 2022, from Feature Selection and Dimensionality Reduction: https://towardsdatascience.com/feature-selection-and-dimensionality-reduction-f488d1a035de

Brownlee, J. (2019). How to Calculate Principal Component Analysis (PCA) from Scratch in Python. Retrieved from https://machinelearningmastery.com/calculate-principal-component-analysis-scratch-python/

Bruce, P. C., Gedeck, P., Shmueli, G., & Patel, N. R. (2019). *Data Mining for Business Analytics Concepts, Techniques and Applications in Python.* Wiley & Sons, Incorporated, John.

Bruce, P., Bruce, A., & Gedeck, P. (2020). *Practical Statistics for Data Scientists 50+ Essential Concepts Using R and Python.* O'Reilly Media, Incorporated.

Fenner, M. (2018). *Machine Learning with Python for Everyone.* Addison Wesley.

Galarnyk, M. (2017). PCA using Python (scikit-learn). Retrieved from https://towardsdatascience.com/pca-using-python-scikit-learn-e653f8989e60

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems.* O'Reilly Media.

h1ros. (2019). Drop Highly Correlated Features. Retrieved from https://h1ros.github.io/posts/drop-highly-correlated-features/

Kaloyanova, E. (2020, March 10). *How to Combine PCA and K-means Clustering in Python?* Retrieved May 8, 2022, from How to Combine PCA and K-means Clustering in Python?: https://365datascience.com/tutorials/python-tutorials/pca-k-means/

Larose, C. D., & Larose, D. T. (2019). *Data Science.* Wiley.

Parveez, S., & Iriondo, R. (2018). Principal Component Analysis (PCA) with Python Examples — Tutorial. Retrieved from https://pub.towardsai.net/principal-component-analysis-pca-with-python-examples-tutorial-67a917bae9aa

ProjectPro. (2022). How to drop out highly correlated features in Python? Retrieved from https://www.projectpro.io/recipes/drop-out-highly-correlated-features-in-python

Sawlani, D. (2017). Customer Churn Prediction Model. Retrieved from https://rstudio-pubs-static.s3.amazonaws.com/344958\_a3c32d38a83043c8ae8fc13cd6abbda4.html

Sharma, A. (2021). PCA or Principal Component Analysis on Customer Churn Data. Retrieved from https://medium.com/data-science-on-customer-churn-data/pca-or-principal-component-analysis-on-customer-churn-data-d18ca60397ed

Srinivas. (2022). Reducing Telecom Churn with PCA and Modeling. Retrieved from https://www.kaggle.com/code/manoharsrinivas/reducing-telecom-churn-with-pca-and-modeling/notebook

VanderPlas, J. (2016). *In Depth: Principal Component Analysis.* O'Reilly. Retrieved from https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysis.html