D212 - Data Mining II - Task 2

Jeremy Dorrough

Western Governors University

WGU Student ID# 000994113

**A1.**

A question that is relevant to a real-world organizational situation that will be addressed using Principal Component Analysis (PCA) is: “Which factors are most significant in determining customer behavior patterns?”

**A2.**

One goal of this data analysis is to identify the primary factors which most impact variance in the customer data to create the best customer segments. Reaching this goal will allow the company to improve overall business performance.

**B1.**

PCA is used to reduce data dimensionality while maintaining important information (Kumar, 2020). It reduces the dataset, transforming the variables into a smaller number of principal components, which are a combination of the original variables and account for variance in the data. An expected outcome of this PCA is that a reduced number of overall data points, represented within the principal components, can be utilized for further analysis. This will allow for better representation and exploration of patterns and data structures, with more straightforward analysis and visualization.

**B2.**

One assumption of PCA is that the principal components are linear combinations as Lever, Krzywinski, and Altman (2017) stated in Nature Methods, *“PCA also has limitations that must be considered when interpreting the output: the underlying structure of the data must be linear.”* This means that PCA assumes a linear relationship between original variables and the resulting principal components. If the relationships between the original variables are nonlinear, PCA may not correctly represent the underlying data structures.

**C1.**

The following continuous dataset variables were used to answer the proposed question in A1:

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

Please note that while Items 1 - 8 in the dataset are in a numerical format, they are being treated as ordinal data and are not included in the analysis.

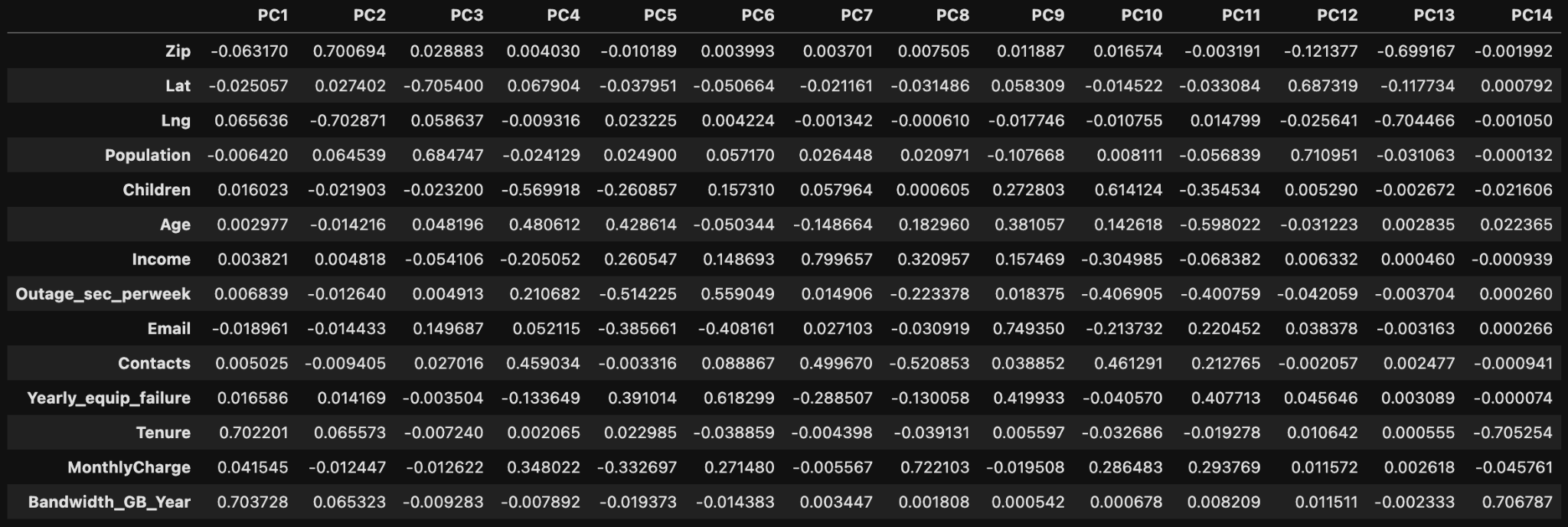
**C2.**

Please see the attached CSV file, “cleaned\_data.csv” for a copy of the cleaned dataset. Please note that the data was standardized using the StandardScaler in the scikit-learn library. For each variable, this subtracted the mean and divided by the standard deviation. Additionally, the code used to perform this, and all following calculations, is included in the “Code Acknowledgement” section of this document.

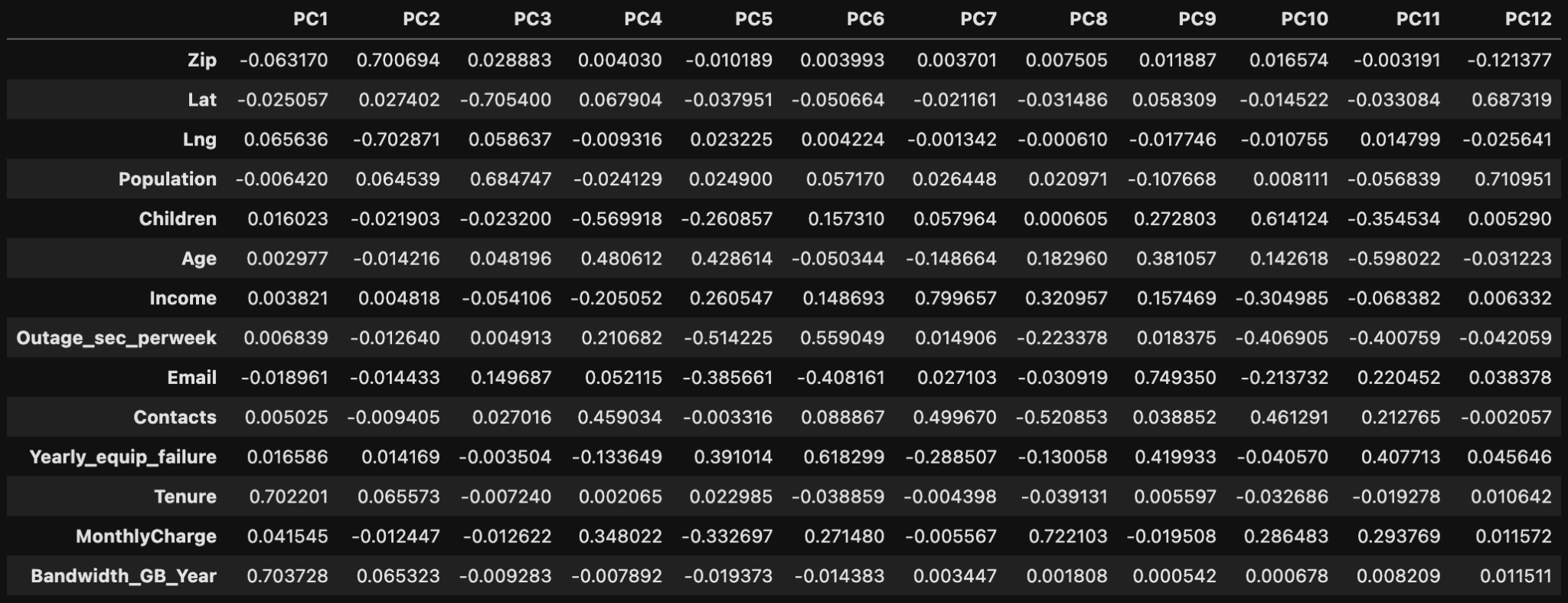
**D1.**

Two principal component matrix tables were generated, before and after utilizing the elbow method to reduce the analysis to the primary 12 principal components. Both the initial and reduced principal component matrix tables can be examined in the screenshots below.

Principal Component Matrix (Before Reduction):

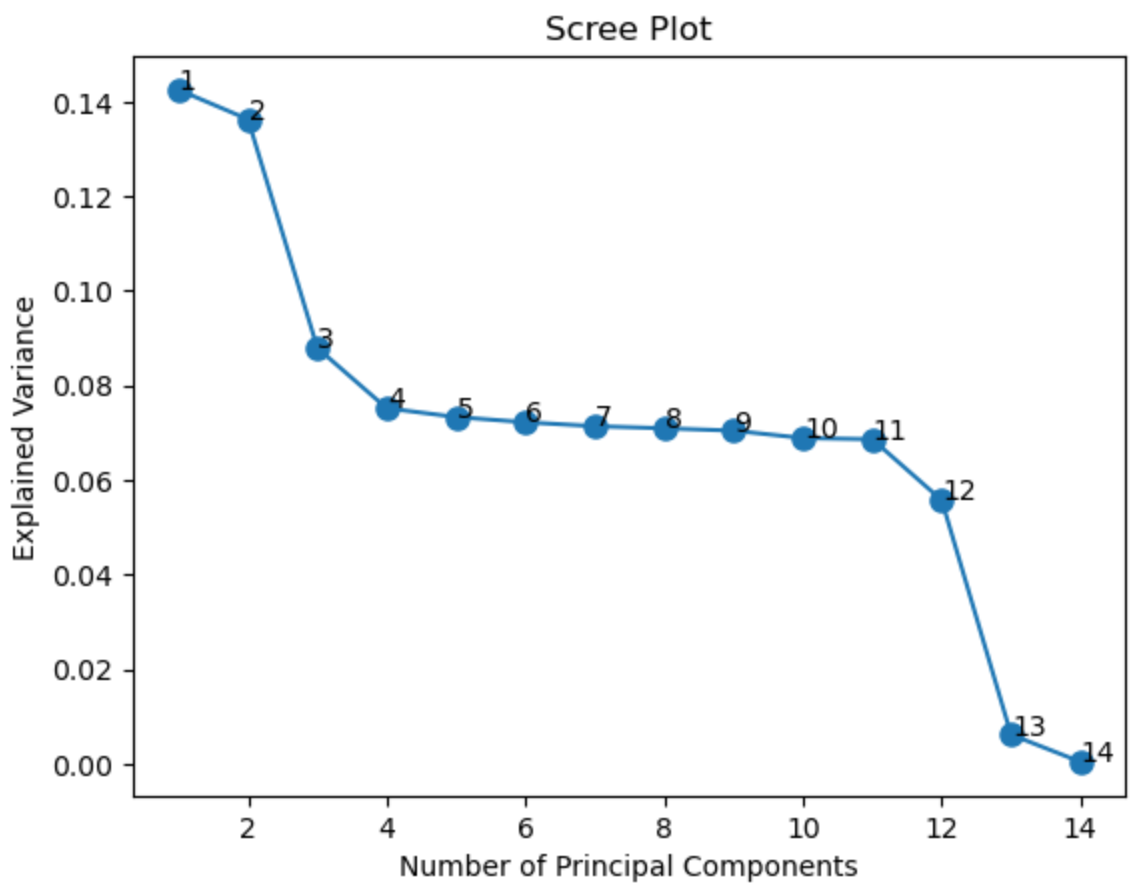


Principal Component Matrix (After Reduction):



**D2.**

Using the elbow rule, a total of 12 principal components were identified. There are two significant elbows, one at the ‘2’ point and one at the ‘12’ point. The latter was selected because the first elbow didn’t represent enough variance, as can be observed in the scree plot below:



**D3.**

Percentage of Variance Explained by Each Principal Component:

* PC1: 14.25%
* PC2: 13.63%
* PC3: 8.79%
* PC4: 7.52%
* PC5: 7.33%
* PC6: 7.22%
* PC7: 7.14%
* PC8: 7.09%
* PC9: 7.04%
* PC10: 6.89%
* PC11: 6.86%
* PC12: 5.57%

**D4.**

Total Variance Explained by the First 12 Principal Components: 99.35%

**D5.**

The purpose of PCA is to reduce dimensionality. However, PCA was only able to reduce the dimensionality of this specific dataset to 12 variables, from an initial 14 variables. There wasn’t a significant reduction in dimensionality, which implies that there is potentially significant redundancy that needs to be addressed or that another technique for data reduction needs to be considered all together.

With that being said, several interesting observations can be made by using the explained variance and matrix results. The Tenure and Bandwidth\_GB\_Year were the highest loadings on PC1 at 0.702201 and 0.703728, respectively. There’s a strong relationship implied between these variables. As PC1 alone explains 14.25% of variance and could provide significant business insights. PC2 and PC3 relate to geographical impact. Zip and Lng are at 0.700694 and -0.702871, respectively in PC2 and Lat and Population are at -0.705400 and 0.684747 in PC3, respectively. This implies that customer patterns are influenced by location and population density and should be further evaluated.

**E.**

Brownee, J. (2020, August 28). How to Use StandardScaler and MinMaxScaler Transforms in Python. Machine Learning Mastery. Retrieved from <https://machinelearningmastery.com/standardscaler-and-minmaxscaler-transforms-in-python/>

Lever, J., Krzywinski, M., & Altman, N. (2017). Principal component analysis. Nature Methods, 14(7), 641-642. <https://doi.org/10.1038/nmeth.4346>

Temlyakov, K. (n.d.). Principal components analysis using pandas dataframe. Stack Overflow. Retrieved from <https://stackoverflow.com/questions/23282130/principal-components-analysis-using-pandas-dataframe>

**F.**

Kumar, S. (2020, August 19). Dimensionality Reduction — Can PCA improve the performance of a classification model. Towards Data Science. Retrieved from <https://towardsdatascience.com/dimensionality-reduction-can-pca-improve-the-performance-of-a-classification-model-d4e34194c544>

**Code Acknowledgment**

The analysis in this paper utilized Python 3.0 and the following libraries:

* pandas: data manipulation/analysis.
* numpy: numerical computing.
* matplotlib: generating plots.
* scikit-learn: machine learning; PCA and data standardization with StandardScaler.

The code used for data analysis is provided below:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.decomposition import PCA

from sklearn.preprocessing import StandardScaler

pd.set\_option('display.max\_columns', None)

Import Data

df = pd.read\_csv('~/Desktop/D212\_Task\_2/churn\_clean.csv', usecols=['Zip','Lat','Lng','Population','Children', 'Age', 'Income', 'Outage\_sec\_perweek', 'Email', 'Contacts', 'Yearly\_equip\_failure', 'Tenure', 'MonthlyCharge', 'Bandwidth\_GB\_Year'])

df.head()

df.shape

#Standardize the continuous dataset variables (Brownee, 2020).

scaler = StandardScaler()

scaled\_data = scaler.fit\_transform(df)

df = pd.DataFrame(scaled\_data, columns=df.columns)

#ExportCSV

df.to\_csv("~/Desktop/D212\_Task\_2/cleaned\_data.csv")

#Perform PCA

pca = PCA()

principal\_components = pca.fit\_transform(df)

#Get loadings matrix

loadings = pca.components\_.T

#Create dataframe with loadings and corresponding features

loadings\_df = pd.DataFrame(loadings, columns=['PC{}'.format(i+1) for i in range(pca.n\_components\_)], index=df.columns)

loadings\_df

#Identify PC count

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

cumulative\_explained\_variance = np.cumsum(explained\_variance)

#Elbow Method

plt.figure()

plt.plot(range(1, len(explained\_variance) + 1), explained\_variance, 'o-', markersize=8)

plt.xlabel('Number of Principal Components')

plt.ylabel('Explained Variance')

plt.title('Scree Plot')

#Number Labels

for i, value in enumerate(explained\_variance):

plt.text(i+1, value, f"{i+1}")

plt.show()

#Assign PC's

num\_principal\_components = 12

#Perform PCA with selected PC's (Temlyakov, n.d.)

pca = PCA(n\_components=num\_principal\_components)

principal\_components = pca.fit\_transform(df)

#Create DF with selected PC's and original index

principal\_components\_df = pd.DataFrame(data=principal\_components,

columns=[f'PC{i+1}' for i in range(num\_principal\_components)],

index=df.index)

#Calculate the percentage of variance explained by each principal component

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

explained\_variance\_percentages = explained\_variance\_ratio \* 100

#Get loadings

loadings = pca.components\_.T

#Create PC Matrix

loadings\_df = pd.DataFrame(loadings, columns=['PC{}'.format(i+1) for i in range(pca.n\_components\_)], index=df.columns)

loadings\_df

#Calculate total variance explained

total\_variance\_explained = np.sum(explained\_variance\_percentages)

#Print variance percentages and total variance explained

print("Percentage of Variance Explained by Each Principal Component:")

for i, percentage in enumerate(explained\_variance\_percentages, start=1):

print(f"PC{i}: {percentage:.2f}%")

print(f"\nTotal Variance Explained by the {num\_principal\_components} Principal Components: {total\_variance\_explained:.2f}%")