Part 1: Research Question

A1) Research Question

The question we will be answering is if we can use PCA as both dimensionality reduction and exploratory analysis to identify which components contribute the most variance in the data.

A2) Define Goal

The goal of this analysis is to use PCA to identify the principal components that contribute the most to understanding the data.

Part 2: Method Justification

B1) Analysis Explanation

PCA analyzes the data by reducing the dimensionality of large data sets into smaller ones while retaining as much information as possible. It creates new uncorrelated variables that successfully maximize the variance. (1) My expected outcome in the analysis is to successfully identify the principal components in the churn data set.

B2) Summary of Assumption

One of the assumptions of using PCA is that there is a linear relationship between the features. (2)

Part 3: Data Preparation

C1) Identifying Variables

To answer the question I proposed, I chose to use most of the continuous variables available in the data set. I decided to exclude the ‘Lat’ and ‘Lng’ columns, as these two do not add much relevant data. All the code is below.

C2) Standardize Variables

To standardize the variables I isolated the variables I wanted to use in a numpy array, then ran the StandardScaler() method before saving the now standardized variables into a new dataframe for the analysis part.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

*import pandas as pd*

*import numpy as np*

*import matplotlib.pyplot as plt*

*import seaborn as sns*

*from sklearn.decomposition import PCA*

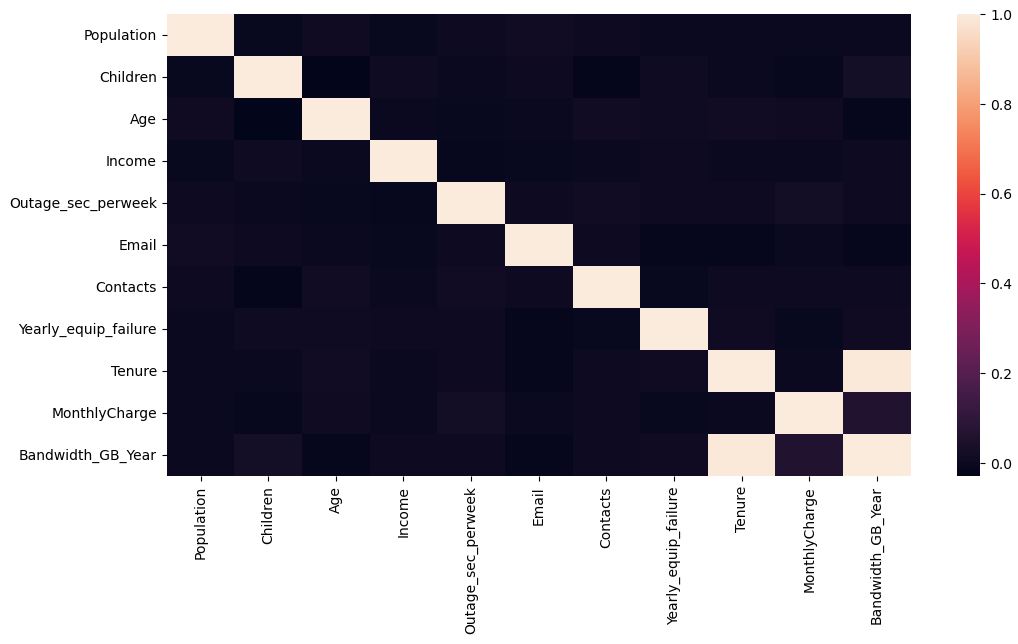
*from sklearn.preprocessing import StandardScaler*

*df = pd.read\_csv('churn\_clean.csv')*

*X = df[['Population', 'Children', 'Age', 'Income', 'Outage\_sec\_perweek', 'Email', 'Contacts', 'Yearly\_equip\_failure', 'Tenure', 'MonthlyCharge', 'Bandwidth\_GB\_Year']].copy()*

*plt.figure(figsize = (12,6))*

*sns.heatmap(X.corr())*

**

*scaled\_var = StandardScaler().fit\_transform(X)*

*scaled\_df = pd.DataFrame(scaled\_var, columns = ['Population', 'Children', 'Age', 'Income', 'Outage\_sec\_perweek', 'Email', 'Contacts', 'Yearly\_equip\_failure', 'Tenure', 'MonthlyCharge', 'Bandwidth\_GB\_Year'])*

*scaled\_df.to\_csv('churn\_prepared.csv', index = False)*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Part 4: Analysis

D1) Determine Matrix

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

*import pandas as pd*

*import numpy as np*

*import matplotlib.pyplot as plt*

*import seaborn as sns*

*from sklearn.decomposition import PCA*

*from sklearn.preprocessing import StandardScaler*

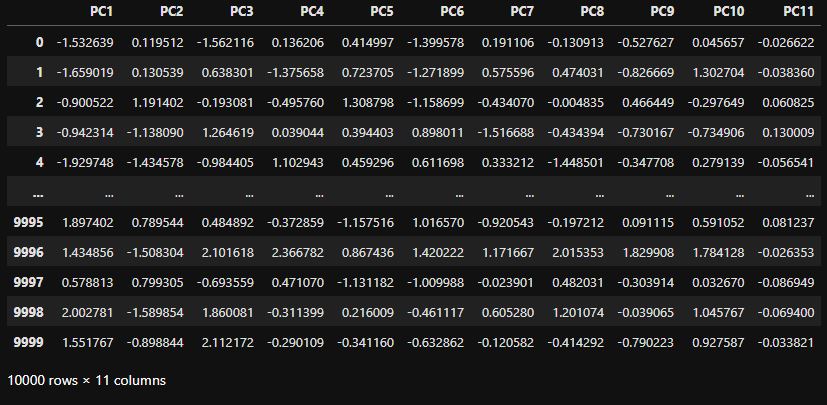
*df = pd.read\_csv('churn\_prepared.csv')*

*model = PCA(n\_components = 11)*

*outcomes = model.fit\_transform(df)*

*loading = pd.DataFrame(outcomes, columns = ['PC1', 'PC2', 'PC3', 'PC4', 'PC5', 'PC6', 'PC7', 'PC8', 'PC9', 'PC10', 'PC11’])*

*loading*



D2) Total Principal Components

Out of the 11 initial components we had, I created an elbow plot to get a better understanding of which principal components should be used. There could be an argument for using either two or ten components, given how the elbow plot looks. Since there is such a sharp decline to the second, I am going to go forward with using the first two PC’s.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

*plt.figure(figsize = (10, 8))*

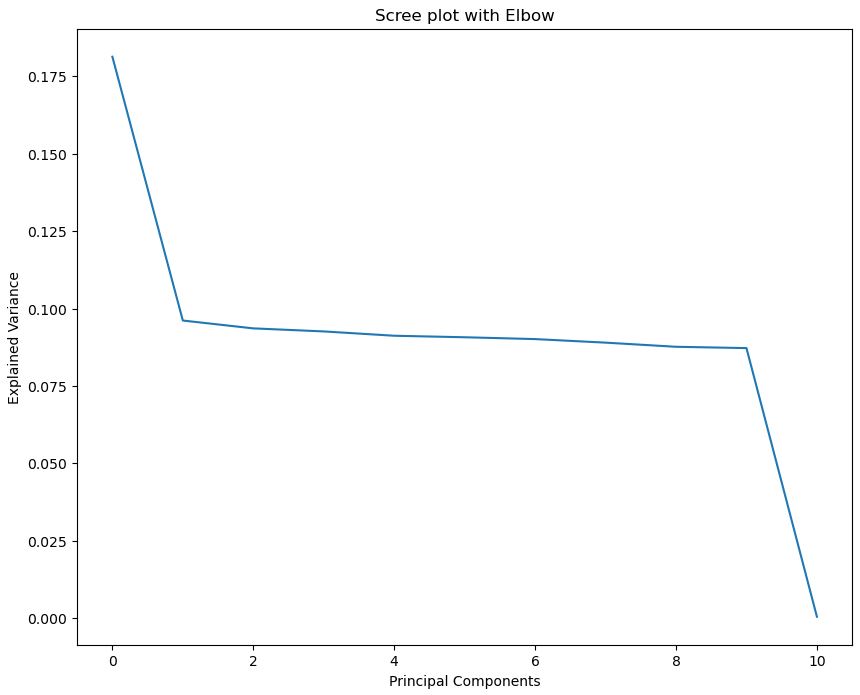
*plt.plot(explained\_ratio)*

*plt.title('Scree plot with Elbow')*

*plt.xlabel('Principal Components')*

*plt.ylabel('Explained Variance')*

*plt.show()*



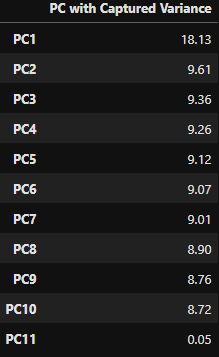
D3) Variance

The following shows the variance for each of the components. Although all are listed, since we are going to keep only the top two for further analysis, the variance for those components are 18.13 and 9.61.

*var = explained\_ratio \* 100*

*Kaiser = pd.DataFrame(var.round(2), columns = ["PC with Captured Variance"], index = ['PC1', 'PC2', 'PC3', 'PC4', 'PC5', 'PC6', 'PC7', 'PC8', 'PC9', 'PC10', 'PC11'])*

*Kaiser*

**

*explained variance = model.explained\_variance\_*

*print("Explained variance for each component: ", explained\_variance)*



D4) Total Variance

When looking at the variance for all the components, we can add the variance for the top two that we want, giving us 27.74. I also have the total found using code below:

*total = Kaiser.loc[['PC1', 'PC2'], 'PC with Captured Variance'].sum()*

*print("The total variance of the two chosen principal components: ", total)*



D5) Summary and Analysis

When setting up for the PCA analysis, I noticed in a correlation map that the variables seem to be all very strongly correlated to each other. Looking above, we see that for the explained variance, most of the components are fairly close together. The first one is a decent amount higher and the last is much lower. Looking at the elbow plot above, we can see that from the first to tenth principal components the variance is steady, with a sharp decline going to the eleventh. When it comes to using PCA to further reduce the dimensionality, an argument could be made for using either two or ten of the components. For this analysis, I used the first two.

F) Sources

1) Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A*, *374*(2065), 20150202. https://doi.org/10.1098/rsta.2015.0202

2) *A Guide to Principal Component Analysis (PCA) for Machine learning*. (n.d.). https://www.keboola.com/blog/pca-machine-learning