Task 2: Dimensionality Reduction (PCA) – D212

Abhishek Aern

Western Governor University

Contents

Part I - Research Question

A1: Question for analysis

How much of the total variance in the churn dataset can be explained by a reduced set of principal components? To what extent can we reduce the dimensionality of the churn dataset while still retaining a significant percentage of the total variance explained by the principal components?

A2: Goal of analysis

The main goal of my analysis is to reduce the number of variables in the dataset while retaining most of the relevant information. To achieve this goal, I will need to transform the original high-dimensional churn dataset into a lower-dimensional space represented by a smaller number of principal components.

This will help me to understand the extent to which the principal components capture the underlying patterns and variability present in the dataset. By quantifying the percentage of variance explained, I can evaluate the effectiveness of the reduced set of components in capturing the key information.

Part II - Method Justification

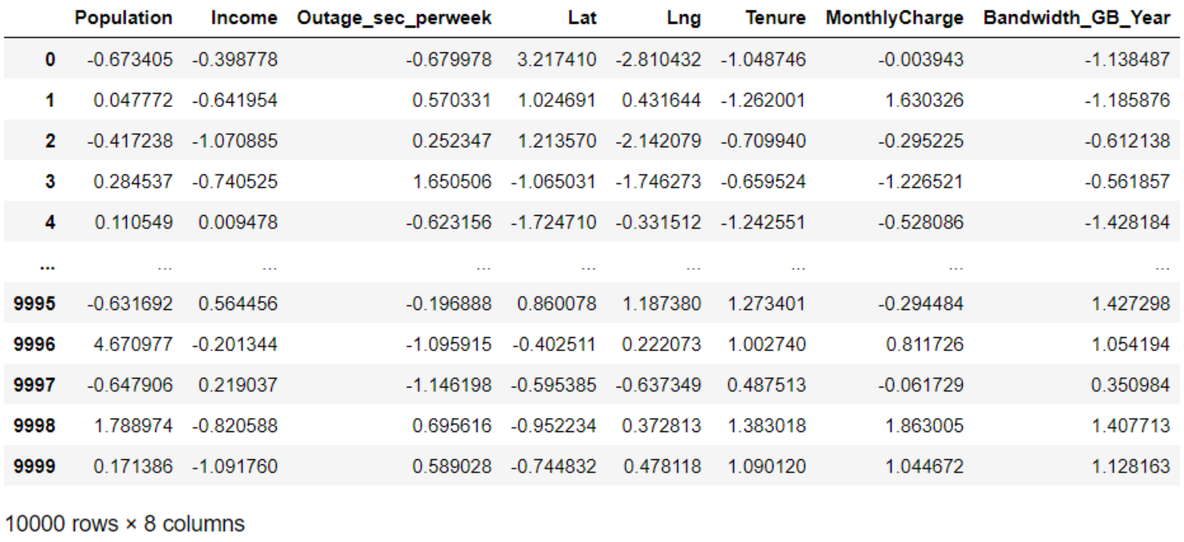
B1: Explanation of PCA

(Turing, n.d.) Principal Component Analysis (PCA) analyzes a selected dataset by transforming the original variables into a new set of uncorrelated variables called principal components. These principal components are linear combinations of the original variables and are ordered in such a way that the first component captures the maximum amount of variance in the data, followed by subsequent components capturing decreasing amounts of variance.

I am selecting all the continuous variables (8 in this case) present in WGU provided telecom company churn dataset and performing the following steps –

Standardization: PCA typically begins by standardizing the dataset, ensuring that each variable has a mean of zero and a standard deviation of one. This will bring all the churn dataset continuous variables on the same scale and gives equal importance to all variables.

Below shows all the variables on the same scale –



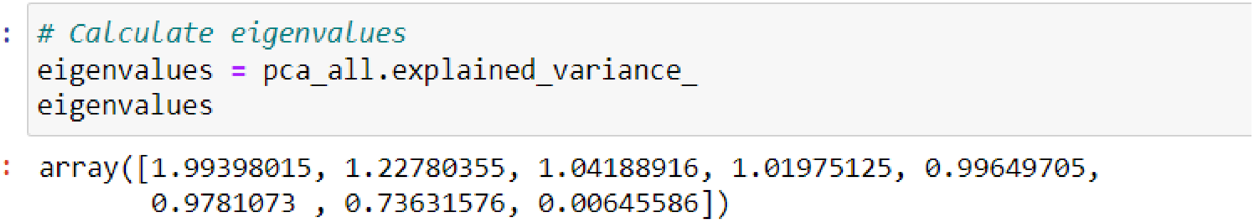
	Population	Income	Outage_sec_perweek	Lat	Lng	Tenure	MonthlyCharge	Bandwidth_GB_Year
0	-0.673405	-0.398778	-0.679978	3.217410	-2.810432	-1.048746	-0.003943	-1.138487
1	0.047772	-0.641954	0.570331	1.024691	0.431644	-1.262001	1.630326	-1.185876
2	-0.417238	-1.070885	0.252347	1.213570	-2.142079	-0.709940	-0.295225	-0.612138
3	0.284537	-0.740525	1.650506	-1.065031	-1.746273	-0.659524	-1.226521	-0.561857
4	0.110549	0.009478	-0.623156	-1.724710	-0.331512	-1.242551	-0.528086	-1.428184
...	...	...	...	...	...	...	...	...
9995	-0.631692	0.564456	-0.196888	0.860078	1.187380	1.273401	-0.294484	1.427298
9996	4.670977	-0.201344	-1.095915	-0.402511	0.222073	1.002740	0.811726	1.054194
9997	-0.647906	0.219037	-1.146198	-0.595385	-0.637349	0.487513	-0.061729	0.350984
9998	1.788974	-0.820588	0.695616	-0.952234	0.372813	1.383018	1.863005	1.407713
9999	0.171386	-1.091760	0.589028	-0.744832	0.478118	1.090120	1.044672	1.128163

10000 rows × 8 columns

Covariance Matrix Calculation: PCA calculates the covariance matrix of the standardized dataset. The covariance matrix provides information about the relationships between pairs of variables, indicating how they vary together. It helps identify which variables are highly correlated and can be combined into fewer principal components. The output is shown in section D1.

Eigenvalue and Eigenvector Calculation: PCA computes the eigenvalues and eigenvectors of the covariance matrix. Each eigenvector represents a principal component, and its corresponding eigenvalue represents the amount of variance explained by that principal component. The eigenvectors are orthogonal to each other, indicating that the principal components are uncorrelated.

Below shows the eigenvalues for each PC from the churn dataset –



```
# Calculate eigenvalues
eigenvalues = pca_all.explained_variance_
eigenvalues
```

```
array([1.99398015, 1.22780355, 1.04188916, 1.01975125, 0.99649705,
       0.9781073 , 0.73631576, 0.00645586])
```

Variance Explained: PCA sorts the eigenvalues in descending order. The eigenvalues represent the amount of variance explained by each principal component. By dividing each eigenvalue by the sum of all eigenvalues to obtain the proportion of variance explained by each component, known as the explained variance ratio. By examining the explained variance ratio, we can identify which principal components contribute the most to the overall variance in the data. The output is shown in section D1.

Selection of Principal Components: I am using the elbow rule for Scree Plot which will help select a subset of principal components based on the desired level of variance explained. The output is shown in section D2. This plot will help us to identify the point where adding more components does not significantly increase the explained variance. The number of components at this point will be considered the optimal number for dimensionality reduction (Turing, n.d.).

B2: Assumptions of PCA

Here are some of the assumptions for PCA (Jain, 2021) –

Linearity: PCA assumes that the relationship between variables is linear. It seeks to capture the linear structure in the data by finding linear combinations of variables that explain the maximum amount of variance.

Independence: PCA assumes that the variables are statistically independent of each other.

Equal Importance of Variables: PCA treats all variables equally in terms of their contribution to the analysis. Variables with larger scales or variances can dominate the principal components, so it is important to standardize the variables before performing PCA.

Part III - Data Preparation

C1: Continuous Dataset Variables

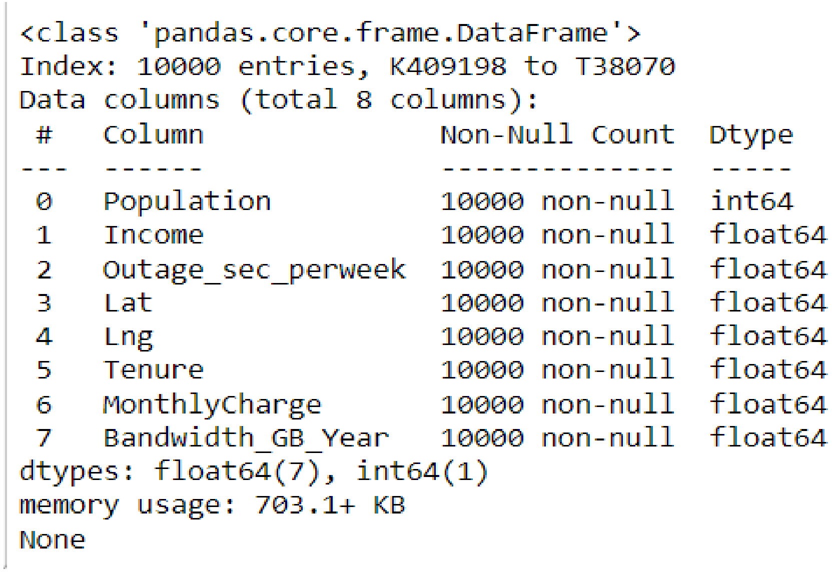
The customer churn dataset has both categorical and continuous variables. I am selecting only continuous variables from the provided churn dataset.

Here are all 8 continuous variables selected for PCA -

Population – Indicating population within a mile radius of the customer.
Income – Numeric value indicating the annual income of the customer.
Outage_sec_perweek - Outage average number of seconds per week.
Lat - Integers indicating GPS coordinates for customers.
Lng - Integers indicating GPS coordinates for customers.
Tenure - Number of years that the customer has been a client with the service provider
MonthlyCharge - The amount customer pays every month to the service provider.
Bandwidth_GB_Year - The average amount of data used by the customer in GB.

The resulting DataFrame df_continuous contains all these columns from the original DataFrame df_churn.

Below is all the information of df_continuous -



```
<class 'pandas.core.frame.DataFrame'>
Index: 10000 entries, K409198 to T38070
Data columns (total 8 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Population          10000 non-null  int64  
 1   Income              10000 non-null  float64
 2   Outage_sec_perweek  10000 non-null  float64
 3   Lat                 10000 non-null  float64
 4   Lng                 10000 non-null  float64
 5   Tenure              10000 non-null  float64
 6   MonthlyCharge       10000 non-null  float64
 7   Bandwidth_GB_Year   10000 non-null  float64
dtypes: float64(7), int64(1)
memory usage: 703.1+ KB
None
```

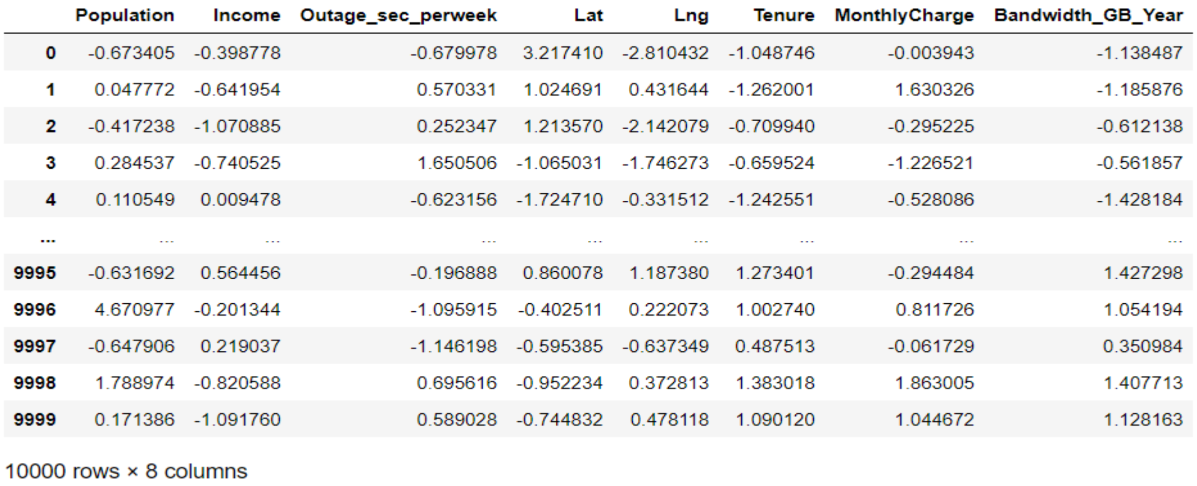
C2: Standardization of Dataset variables

Standardizing the variables is important in PCA to ensure that each variable contributes equally and is on the same scale, avoiding any bias introduced by differences in variable units or ranges. I am using StandardScaler class from scikit-learn to Standardize the continuous dataset variables identified as part of C1.

First, an instance of StandardScaler is created. Then, the scaler is fitted on the continuous dataset using the fit() method, which calculates and stores the mean and standard deviation of each feature.

Next, the standardized transformation is applied to the dataset using the transform() method. This scales the variables by subtracting the mean and dividing them by the standard deviation, resulting in a dataset with a mean of zero and a standard deviation of one for each variable.

Finally, the resulting standardized array is converted back to a DataFrame using pd.DataFrame(), with the column names preserved from the original df_continuous DataFrame. The df_standardized DataFrame now contains the standardized continuous variables.



	Population	Income	Outage_sec_perweek	Lat	Lng	Tenure	MonthlyCharge	Bandwidth_GB_Year
0	-0.673405	-0.398778	-0.679978	3.217410	-2.810432	-1.048746	-0.003943	-1.138487
1	0.047772	-0.641954	0.570331	1.024691	0.431644	-1.262001	1.630326	-1.185876
2	-0.417238	-1.070885	0.252347	1.213570	-2.142079	-0.709940	-0.295225	-0.612138
3	0.284537	-0.740525	1.650506	-1.065031	-1.746273	-0.659524	-1.226521	-0.561857
4	0.110549	0.009478	-0.623156	-1.724710	-0.331512	-1.242551	-0.528086	-1.428184
...	...	...	...	...	...	...	...	...
9995	-0.631692	0.564456	-0.196888	0.860078	1.187380	1.273401	-0.294484	1.427298
9996	4.670977	-0.201344	-1.095915	-0.402511	0.222073	1.002740	0.811726	1.054194
9997	-0.647906	0.219037	-1.146198	-0.595385	-0.637349	0.487513	-0.061729	0.350984
9998	1.788974	-0.820588	0.695616	-0.952234	0.372813	1.383018	1.863005	1.407713
9999	0.171386	-1.091760	0.589028	-0.744832	0.478118	1.090120	1.044672	1.128163

10000 rows × 8 columns

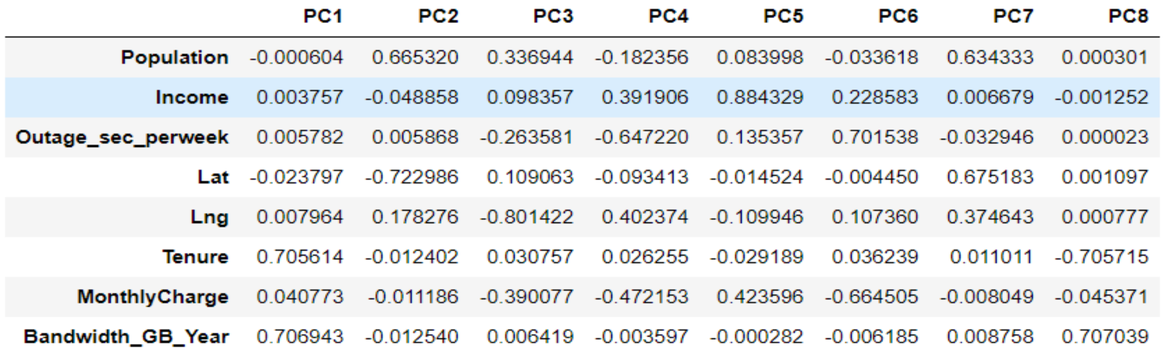
The prepared clean data is provided in a CSV file.

Part IV - Analysis

D1: Principal Components

To apply the PCA, I created an instance of the PCA class and fit it on the standardized feature matrix. This step calculates the principal components and their corresponding eigenvalues.

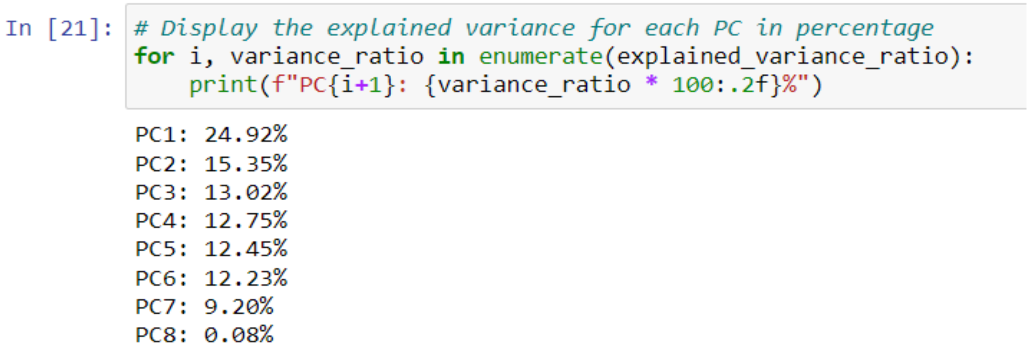
PCA loading is shown below for all the components. By examining these loadings, we can identify the variables that contribute the most to each principal component.



	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
Population	-0.000604	0.665320	0.336944	-0.182356	0.083998	-0.033618	0.634333	0.000301
Income	0.003757	-0.048858	0.098357	0.391906	0.884329	0.228583	0.006679	-0.001252
Outage_sec_perweek	0.005782	0.005868	-0.263581	-0.647220	0.135357	0.701538	-0.032946	0.000023
Lat	-0.023797	-0.722986	0.109063	-0.093413	-0.014524	-0.004450	0.675183	0.001097
Lng	0.007964	0.178276	-0.801422	0.402374	-0.109946	0.107360	0.374643	0.000777
Tenure	0.705614	-0.012402	0.030757	0.026255	-0.029189	0.036239	0.011011	-0.705715
MonthlyCharge	0.040773	-0.011186	-0.390077	-0.472153	0.423596	-0.664505	-0.008049	-0.045371
Bandwidth_GB_Year	0.706943	-0.012540	0.006419	-0.003597	-0.000282	-0.006185	0.008758	0.707039

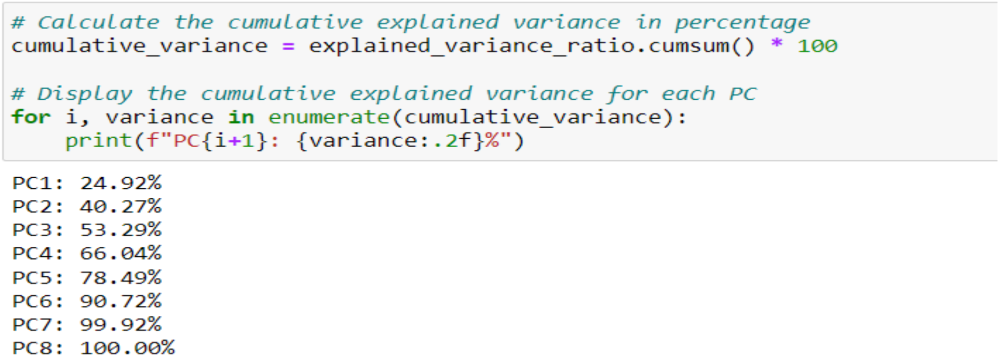
Loadings are interpreted as the weights or coefficients that determine the influence of each variable on the principal components. Positive loadings indicate a positive relationship between the variable and the component, while negative loadings indicate a negative relationship. The variables with higher absolute loading values have a stronger influence on the respective principal component. The Interpretation of this matrix is described in section D5.

To understand how much of the total variance is explained by each principal component I am using the explained_variance_ratio_ attribute on the pca_all object.



```
In [21]: # Display the explained variance for each PC in percentage
         for i, variance_ratio in enumerate(explained_variance_ratio):
             print(f"PC{i+1}: {variance_ratio * 100:.2f}%")

         PC1: 24.92%
         PC2: 15.35%
         PC3: 13.02%
         PC4: 12.75%
         PC5: 12.45%
         PC6: 12.23%
         PC7: 9.20%
         PC8: 0.08%
```

Below shows the cumulative explained variance in percentage for all the PCs –



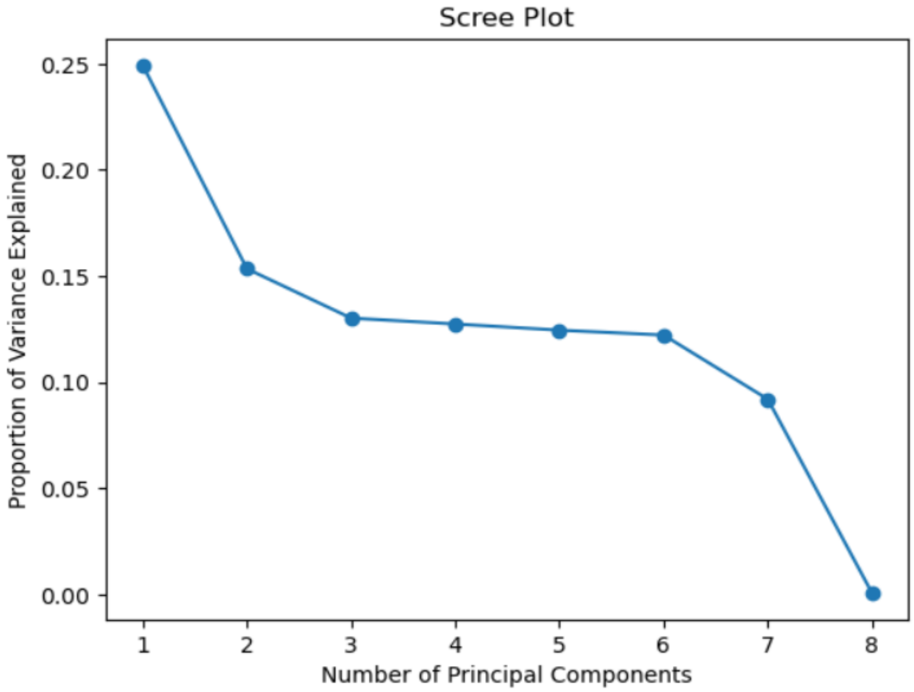
```
# Calculate the cumulative explained variance in percentage
cumulative_variance = explained_variance_ratio.cumsum() * 100

# Display the cumulative explained variance for each PC
for i, variance in enumerate(cumulative_variance):
    print(f"PC{i+1}: {variance:.2f}%")

PC1: 24.92%
PC2: 40.27%
PC3: 53.29%
PC4: 66.04%
PC5: 78.49%
PC6: 90.72%
PC7: 99.92%
PC8: 100.00%
```

I am computing the cumulative explained variance by applying the cumsum() function to the explained variance ratio array and multiplying it by 100 to obtain the percentage. By examining the cumulative explained variance, we can determine how many principal components are needed to capture a certain percentage of the total variance. For instance, if I need to explain at least 90% of the variance, then need to retain the first 6 principal components (PC1 to PC6), as they collectively explain around 91% of the total variance.

D2: Identification of the total number of Components

(Naik, 2023) The elbow rule and the Kaiser criterion are two common methods to identify the total number of principal components to retain in a PCA analysis.

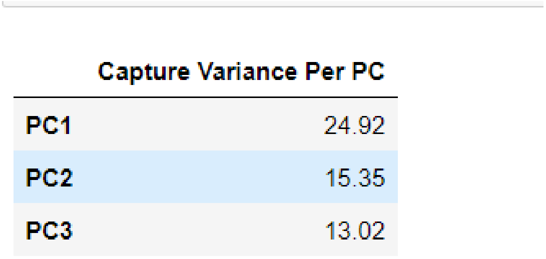
I am using the Elbow Rule which involves plotting the explained variance against the number of principal components and looking for a point of inflection, often resembling an elbow shape. The idea is to select the number of components at the elbow point, as it represents a trade-off between capturing enough variance and avoiding overfitting.



Based on the scree plot, the first three principal components capture a significant portion of the total variance in your dataset. PC1, PC2, and PC3 likely have higher explained variance ratios compared to the remaining components so this can be retained for further analysis.

D3: Total Variance of Components

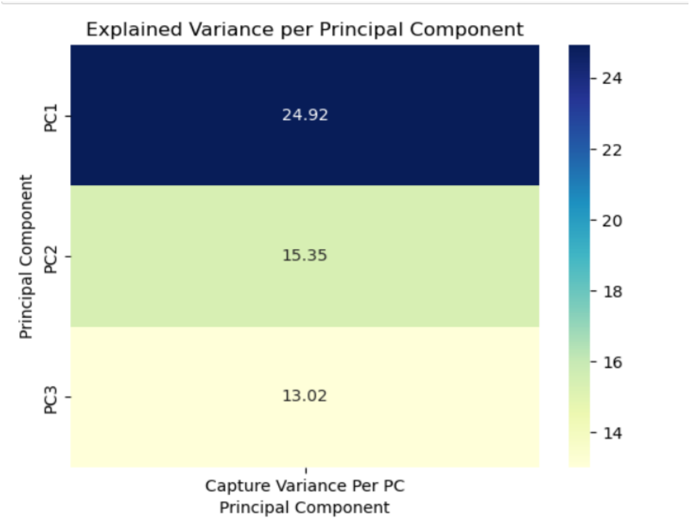
Based on the above-selected PCs I am again applying PCA with n_components=3. Here is the captured variance per PC for the selected first 3 components in percentage. This is calculated using explained variance ratio.



	Capture Variance Per PC
PC1	24.92
PC2	15.35
PC3	13.02

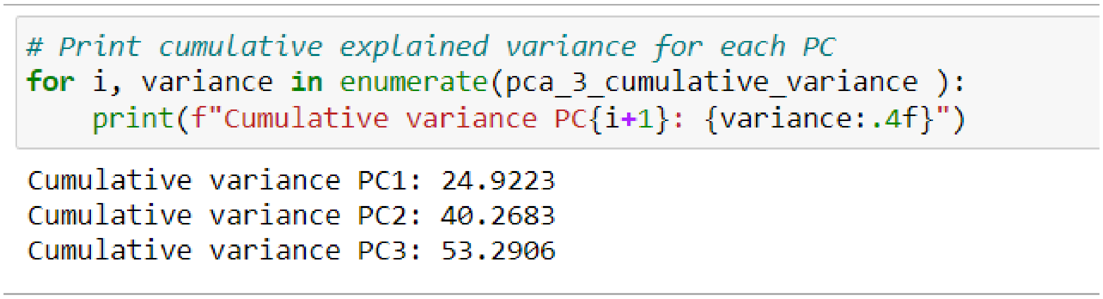
This indicates that PC1 is the most influential component in explaining the variability(24.92% of the total variance) in the dataset. PC2 contributes significantly to explaining the variability in the data, but to a lesser extent than PC1. PC3 contributes further to explaining the variability in the data, but again with a smaller impact compared to PC1 and PC2. In total, they all combined explain 53.29% variability in the dataset.

Below is the heat map for the same.



D4: Total Variance captured by Components

Below cumulative variance percentages indicate the cumulative amount of variance explained by each principal component in a stepwise manner.



```
# Print cumulative explained variance for each PC
for i, variance in enumerate(pca_3_cumulative_variance ):
    print(f"Cumulative variance PC{i+1}: {variance:.4f}")
```

```
Cumulative variance PC1: 24.9223
Cumulative variance PC2: 40.2683
Cumulative variance PC3: 53.2906
```

Each principal component captures a certain amount of variance, and the cumulative variance reflects the total amount of variance explained by including the current and previous components. Like PC1 explains 24.92% of the total variance in the dataset, PC2 explains an additional 15.34% of the total variance, and so on. By selecting all these 3 PCs we can explain around 53.29% of the variance of the telecom company churn dataset.

D5: Summary of Data Analysis

Below is the summary based on all the above analysis -

- Each retained principal component (PC1, PC2, and PC3) represents a linear combination of the original variables.

- Using the elbow method 8 principal components are reduced to 3 PCs as components beyond the elbow point not contributing significantly to the overall variance explained.

- The loading matrix provides information about the contributions of each original variable to each principal component. Based on the loading matrix –

    - PC1 is strongly influenced by the "Tenure" and "Bandwidth_GB_Year" variables, as they have high absolute loading values. It suggests that customers with longer tenure and higher bandwidth usage contribute significantly to this component.

- PC2 is mainly influenced by the "Lat", "Lng", and "Population" variables, indicating a relationship between geographic location and population density.

- PC3 indicates the "Income" variable has a relatively high loading in this component, suggesting that it plays a crucial role in explaining the variation in this component.

- PC4 indicates the "Income" and "Outage_sec_perweek" variables have notable loadings in this component, indicating a potential relationship between income and service outage duration.

Based on these interpretations, we get all the insights into the relationships between the variables and the principal components and which will help in the model-building process to improve model interpretability, and potentially enhance model performance by removing irrelevant or redundant variables.

E. Third-Party Code References

WGU, Resources. (n.d.). pca_webinar_recording.
https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=60fa4159-94ba-4f41-9ba8-aea0011f18f9

Naik, K. (2023, Jan 18). PCA Indepth Geometric And Mathematical InDepth Intuition ML Algorithms.
https://www.youtube.com/watch?v=H99JRtDDnvk

F. References

Jain, S. (2021, May 15). Limitations, Assumptions Watch-Outs of Principal Component Analysis.
https://codatalicious.medium.com/limitations-assumptions-watch-outs-of-principal-component-analysis-8483ceaa2800

Turing. (n.d.). A Step-By-Step Complete Guide to Principal Component Analysis | PCA for Beginners.
https://www.turing.com/kb/guide-to-principal-component-analysis