D212 Task 2: Dimensionality Reduction

Part 1: Research Question

1. Which features in the data set are the most important? What is the best number of features to use for analysis without loss of information?

2. This analysis aims to reduce the number of features to only the most important. Reducing the number of features will help avoid overfitting. Feature reduction also allows for the generalization of models. A small number of variables also reduces the computation power needed to run the model. (Goonewardana, 2019)

Part 2: Method Justification

1. Principal component analysis (PCA) is used to reduce the dimensionality of a data set. It is a transformation that changes a set of possibly correlated features into uncorrelated features called principal components. These are the steps for PCA:

- Standardize the data
- Calculate the covariance matrix
- Find the eigenvectors and the eigenvalue
- Sort the principal components based on their eigenvalues and create a scree plot to choose the number of eigenvectors (k) to keep. The number of components retained is based on the explained variance the components have.
- Create the projection matrix using the selected k and transform the original data set.

  (A Step-By-Step Introduction to Principal Component Analysis (PCA) with Python, 2020)

2. One assumption of PCA is a linear relationship between all features. Because PCA is based on Pearson correlation coefficients, there must be a linear relationship between the variables. (Principal Components Analysis (PCA) using SPSS Statistics, n.d.)

Part 3: Data Preparation

1. The continuous data set features used to answer this question are Population, Children, Age, Income, Outage_sec_perweek, Email, Contacts, Yearly_equip_failure, Tenure, MonthlyCharge, Bandwidth_GB_Year.

2. Copy of cleaned data attached as: PCA_Prepared_Data.csv

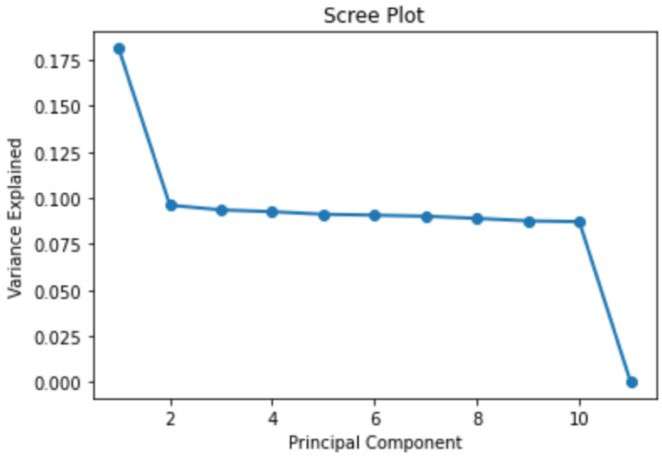
Part 4: Analysis

1. After the data was fit and transformed using PCA, the following principal component matrix was determined:

	PC 1	PC 2	PC 3	PC 4	PC 5	PC 6	PC 7	PC 8	PC 9	PC 10	PC 11
0	-1.532639	0.119512	-1.562116	0.136206	0.414997	-1.399578	0.191106	-0.130913	-0.527627	0.045657	-0.026622
1	-1.659019	0.130539	0.638301	-1.375658	0.723705	-1.271899	0.575596	0.474031	-0.826669	1.302704	-0.038360
2	-0.900522	1.191402	-0.193081	-0.495760	1.308798	-1.158699	-0.434070	-0.004835	0.466449	-0.297649	0.060825
3	-0.942314	-1.138090	1.264619	0.039044	0.394403	0.898011	-1.516688	-0.434394	-0.730167	-0.734906	0.130009
4	-1.929748	-1.434578	-0.984405	1.102943	0.459296	0.611698	0.333212	-1.448501	-0.347708	0.279139	-0.056541
...	...	...	...	...	...	...	...	...	...	...	...
9995	1.897402	0.789544	0.484892	-0.372859	-1.157516	1.016570	-0.920543	-0.197212	0.091115	0.591052	0.081237
9996	1.434856	-1.508304	2.101618	2.366782	0.867436	1.420222	1.171667	2.015353	1.829908	1.784128	-0.026353
9997	0.578813	0.799305	-0.693559	0.471070	-1.131182	-1.009988	-0.023901	0.482031	-0.303914	0.032670	-0.086949
9998	2.002781	-1.589854	1.860081	-0.311399	0.216009	-0.461117	0.605280	1.201074	-0.039065	1.045767	-0.069400
9999	1.551767	-0.898844	2.112172	-0.290109	-0.341160	-0.632862	-0.120582	-0.414292	-0.790223	0.927587	-0.033821

[10000 rows x 11 columns]

2. A scree plot was created to identify the principal components to keep.

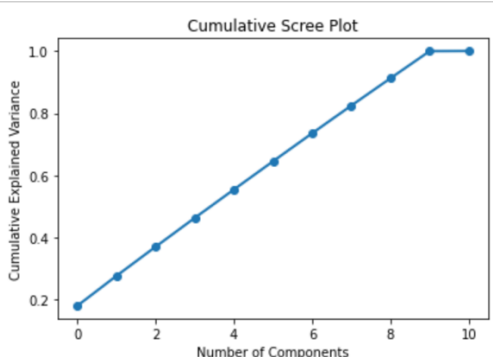


Using the elbow method, the number of principal components to keep is 10.

3. The variance for each principal component is listed in the following table:

Component	1	2	3	4	5	6	7	8	9	10	11
Variance	.1812	.0961	.0935	.0926	.0912	.0907	.0901	.0889	.0876	.0872	.0005

4. The total variance captured by the ten principal components kept is 99.95%.



Component	1	2	3	4	5	6	7	8	9	10
Variance	.1812	.2774	.3709	.4636	.5548	.6455	.7356	.8246	.9122	.9995

5. PCA was used to reduce the data set to a more manageable size. This reduction has also helped to avoid overfitting because there are fewer variables. The analysis shows that only ten components are needed with very little loss of information.

Works Cited

A Step-By-Step Introduction to Principal Component Analysis (PCA) with Python. (2020, April 25). Retrieved December 2021, from Data Science Samurai: https://datasciencesamurai.com/step-by-step-principal-component-analysis-pca-python/

Goonewardana, H. (2019, February 28). *PCA: Application in Machine Learning*. Retrieved December 2021, from Apprentice Journal: https://medium.com/apprentice-journal/pca-application-in-machine-learning-4827c07a61db

Principal Components Analysis (PCA) using SPSS Statistics. (n.d.). Retrieved December 2021, from Laerd Statistics: https://statistics.laerd.com/spss-tutorials/principal-components-analysis-pca-using-spss-statistics.php