D212: Data Mining II

Principal Component Analysis of Telecom Customer Dataset

Andrew Mecchi

Masters of Data Analytics, Western Governors University

The following report covers Task 2 of the Performance Assessment for D212 Data Mining II. This document is categorized by the questions defined in the rubric.

Part I: Research Question

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

As an analyst for WGU Telecom it is essential to understand customer habits and tendencies. The data dictionary informs that customer churn rates may be as high as 25% in the telecommunications industry, therefore, recognizing customer characteristics through principal component analysis could lead to better business strategies and decision-making. However, the provided dataset has high dimensionality with 50 features which makes extracting important information about customers extremely difficult.

Can principal component analysis be used to identify important features that contribute to common characteristics of WGU Telecom subscribers?

A2. Define one goal of the principal component analysis.

The goal of this analysis is to successfully reduce the dimensionality of the WGU Telecom dataset. The results of an effective principal component analysis allows for subsequent analysis of the targeted principal components and will assist with further assessment of customer habits which will lead to better, informed decision-making.

Part II: Method Justification

B1. Explain how PCA analyzes the selected data set. Include expected outcomes.

Principal component analysis (PCA) is an efficient tool used to assess large data sets and expunge important information from a large number of variables that would otherwise be very time-consuming. Principal component analysis is an unsupervised type of feature extraction where original variables are combined and reduced to their most important and descriptive components without compromising important traits (Firmin, 2019). PCA is a process that involves standardization of variables, calculation of a covariance matrix, computation of eigenvectors and eigenvalues, principal component identification, and principal component reduction determined by Kaiser criterion.

The first step of PCA is to standardize the imputed data so each feature contributes to the analysis equally. The role of the covariance matrix is to summarize the correlations between all possible pairs of features by computing how the variables vary from the mean with respect to each other (Pierre, 2020). The values from the covariance matrix assist with the calculation of eigenvectors and eigenvalues. Eigenvectors show how much each input variable contributes to each newly derived principal component and eigenvalues represent the proportion of variance

explained by each principal component (Great Learning Team, 2022). The principal components are new variables determined from linear transformations of the original imputed data, whereby eigenvectors calculate weights to be used in the transformation and eigenvalues inform the magnitude of the explained variance (Schork, n.d.). Dimensionality reduction takes its final form when the Kaiser criterion is enforced to determine the most significant principal components. The Kaiser criterion retains components with eigenvalues greater than one and drops those with eigenvalues less than or equal to one.

For the WGU Telecom principal component analysis, the process starts with isolating all continuous variables from the original dataset. A standard scaler is then applied to those continuous features which sets mean values to zero with a standard deviation of one. A new dataframe is created with the number of defined principal components equal to the number of variables input into the analysis. The PCA function is applied to the scaled data and has the measure of the corresponding explained_variance_ attribute entered for each respective principal component value. The explained_variance_ attribute is as self-described, it is the quantifiable variance explained by each principal component, or, the eigenvalue. Further analysis of the eigenvalue is done and the Kaiser criterion is employed, dropping components that don't exceed the threshold of one. Therefore, the expected outcome of the PCA will remove principal components with eigenvalues of one or less while components with values greater than one are retained. Ultimately, the overall dimensionality of the dataset is greatly reduced with the removal of feature components with low explained variance while keeping the statistically significant principal components.

B2. Summarize one assumption of PCA.

When performing dimensionality reduction through principal component analysis, it is important to understand if the data qualifies for this method of reduction based on the operational assumptions of PCA. One such assumption is to understand that PCA is sensitive to the magnitude and scale of the features (Keboola, n.d.). Principal component analysis attempts to maximize explained variance of features, thus scaling features is paramount. Often, features included in a PCA are measured in different units. For example, variables from the WGU Telecom dataset; Bandwidth is quantified in gigabytes per year while Outages are recorded in seconds per week. Bandwidth has values (approximately) between 155-7159 and values of Outages (sec per week) measure from 0-22 (approximate). Therefore, if PCA was performed on these two features without scaling, the analysis will be extremely biased to the variance of the feature with a wider range compared to the smaller range. The results of these unscaled features in this PCA example would heavily skew the first principal component to the larger scaled feature (Bandwidth). To address this feature bias of unscaled variables, a standard scaler is used on the dataset to allow for each variable to equally contribute to the overall explained variance.

Part III: Data Preparation

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

Feature	Data Type
Income	Continuous
Outage_sec_per_week	Continuous
Tenure	Continuous
MonthlyCharge	Continuous
Bandwidth_GB_Year	Continuous

Note: Previous to running principal component analysis, the dataframe created to isolate continuous features originally included Latitude and Longitude. While these two features qualify as continuous variables, they were dropped from the dataframe as latitude and longitude correspond to a location within a city or town, otherwise, categorical data in nature. Therefore, the features that meet the continuous quantitative variable requirements for principal component analysis are Income, Outages (seconds per week), Tenure, Monthly Charges, and Bandwidth (gigabytes per year).

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

1) Isolate continuous quantitative variables from original dataframe

```
# PCA dataframe creation
# Search for CONTINUOUS numerical data only
t2 = pd.DataFrame(data.select_dtypes(include = 'float64'))
```

2) Drop latitude and longitude from dataframe, only continuous quantitative variables remain

```
# Create df with only continuous quantitative variables - drop lat and lng
df = t2.drop(columns = ['Lat', 'Lng'])
```

3) Prepare standard scaler object

```
# Instantiate Standard Scaler
scaler = StandardScaler()
```

4) Fit scaler to continuous variable dataframe

```
# Fit scaler to dataset
scaler.fit(df)
```

5) Apply standard scaler transforming feature values

```
# Scale samples to transform values
scaled = scaler.transform(df)
```

6) Provide copy of scaled data for PA

```
# Export copy of scaled data in a dataframe
df_scaled = pd.DataFrame(scaled, index = df.index, columns = df.columns)

# Export df for PA
df_scaled.to_csv(r'C:\\Users\\andrew\\Desktop\\WGU_MSDA\\D212_Data_Mining_
II\\PA\\Task_2\\df_scaled_for_pca.csv')
```

7) Confirmation of scaled features

```
# Confirm scaled data
scaled_summary_stats = df_scaled.describe()

# Observe mean of 0 and standard deviation  of 1
scaled_summary_stats.round(1)
```

Figure 1: Confirmation of Scaled Data for PCA - Mean = 0, StDev = 1

	Income	Outage_sec_perweek	Tenure	MonthlyCharge	Bandwidth_GB_Year
count	10000.0	10000.0	10000.0	10000.0	10000.0
mean	0.0	0.0	0.0	-0.0	0.0
std	1.0	1.0	1.0	1.0	1.0
min	-1.4	-3.3	-1.3	-2.2	-1.5
25%	-0.7	-0.7	-1.0	-0.8	-1.0
50%	-0.2	0.0	0.0	-0.1	-0.1
75%	0.5	0.7	1.0	0.7	1.0
max	7.8	3.8	1.4	2.7	1.7

8) Copy of cleaned dataset

SEE ATTACHED: df_scaled_for_pca.csv

Part IV: Analysis

D1. Determine the matrix of *all* the principal components.

The continuous variables have been identified, scaled with the standard scalar, and are ready for principal component analysis. First, the PCA object is instantiated with n_components equal to the number of variables input in the analysis (5) and a random_state is added for reproducibility. Once the PCA object is instantiated, it is fit to the scaled samples, transformed to apply dimensionality reduction, and the results are entered into a new dataframe (Fig. 2). The matrix dataframe is created to examine the PCA loadings which identify the weight each feature contributes to the five principal components (Fig. 3). The values of the loadings represent the direction of maximum variance in the data and sort the components in decreasing order of explained variance (Scikit-learn, n.d). Each column correlates to a principal component while the values in the rows quantify the weight of the corresponding variable; with higher values (closer to 1 or -1) carrying the most influence for that component. For example, principal component one (PC_1) is predominantly influenced by Tenure and Bandwidth (gigabytes per year) while the remaining variables carry little weight relative to PC_1 (Fig. 4).

Figure 2: Principal Component Analysis - Fit and Transform Scaled Data

```
# Instantiate PCA with all features in df
pca = PCA(n_components = df.shape[1], random_state = 61)

pca.fit(scaled)
```

```
PCA(n_components=5, random_state=61)
```

```
[129] # Fit and transform  pca to scaled data - scaled
      pc = pca.transform(scaled)
```

```
[134] # Create new df with fit and transformed pc's
      df_pc = pd.DataFrame(pc, columns = ['PC_1,', 'PC_2', 'PC_3', 'PC_4', 'PC_5'])

      # View first three rows of new pc df
      df_pc.head(3)
```

	PC_1,	PC_2	PC_3	PC_4	PC_5
0	-1.551056	0.279016	-0.380379	-0.620483	-0.064167
1	-1.662133	-1.703695	0.213964	-0.826325	-0.020960
2	-0.948752	-0.406056	-1.047420	0.085881	0.082959

Figure 3: View of Principal Component Analysis Loadings

```
[135] # Create loadings df
      loadings = pd.DataFrame(pca.components_.T, columns = ['PC_1', 'PC_2', 'PC_3', 'PC_4', 'PC_5'], index = df.columns)

      # View principal component loadings
      loadings
```

	PC_1	PC_2	PC_3	PC_4	PC_5
Income	0.003936	0.365683	0.897256	0.247368	-0.001250
Outage_sec_perweek	0.005858	-0.682929	0.078077	0.726276	0.000034
Tenure	0.705844	0.039973	-0.030063	0.035159	-0.705724
MonthlyCharge	0.040748	-0.631099	0.433512	-0.640363	-0.045361
Bandwidth_GB_Year	0.707159	0.000089	-0.000613	-0.005588	0.707032

Figure 4: PC 1 Feature Influence

	PC_1
Income	0.003936
Outage_sec_perweek	0.005858
Tenure	0.705844
MonthlyCharge	0.040748
Bandwidth_GB_Year	0.707159

D2. Identify the *total* number of principal components using the elbow rule or the Kaiser criterion. Include a screenshot of the scree plot.

Following completion of the principal component analysis, the total number of principal components to retain is two. The number of principal components to keep for dimensionality reduction is determined by enforcing the Kaiser criterion, whereby eigenvalues greater than one are kept while the remaining components are discarded. Eigenvalues from the PCA are calculated by applying the explained_variance_ attribute which totals the explained variance of each feature. When observing the eigenvalues in their respective results dataframe (Fig. 5) and scree plot (Fig. 6), there are two principal components that have eigenvalues greater than one and are thus retained.

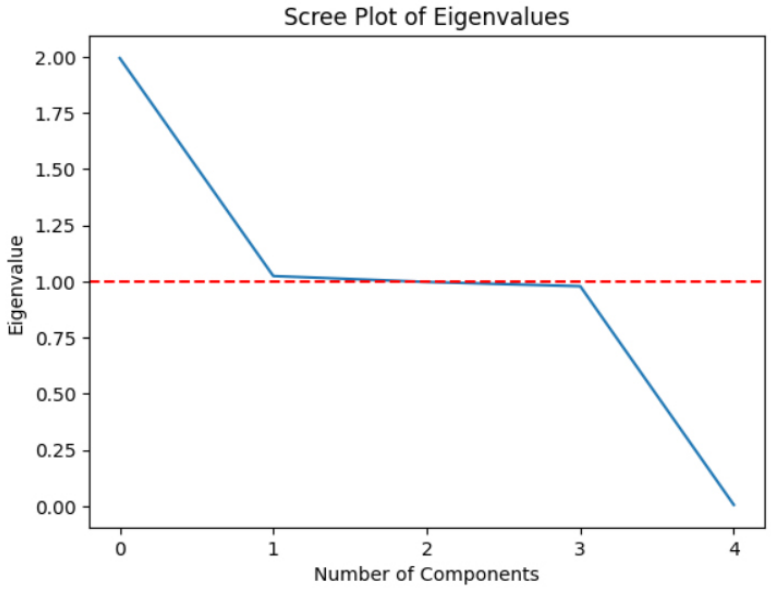
Figure 5: Eigenvalues from Principal Component Analysis

```
[141] # Eigenvalues
      # Calculate Eigenvalues
      eigenvalues = pca.explained_variance_

      # Create df with eigenvalues and pc's
      df_eigenvalues = pd.DataFrame(eigenvalues.round(3), columns = ['Eigenvalue per PC'], index = ['PC_1', 'PC_2', 'PC_3', 'PC_4', 'PC_5'])

      # View eigenvalue df
      df_eigenvalues
```

	Eigenvalue per PC
PC_1	1.993
PC_2	1.024
PC_3	0.998
PC_4	0.979
PC_5	0.006

Principal Components 1 & 2 are retained based on Kaiser criterion: eigenvalue > 1

Figure 6: Scree Plot with Line-Marker for Kaiser Criterion



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

Principal Component	Explained Variance per PC (eigenvalue)
PC 1	1.993
PC 2	1.024
PC 3	0.998
PC 4	0.979
PC 5	0.006

D4. Identify the *total* variance captured by the principal components identified in part D2.

The total variance captured by the two retained principal components is 60.34%, which is calculated by adding the totals of the percentage of variance explained by the first two components (Fig. 7). The percent variance is calculated by multiplying the explained_variance_ratio_ attribute by one-hundred, converting the ratio into a percentage. The total variance captured by all principal components can be visualized with a heatmap (Fig. 8) and additionally supported by a scree plot representing the cumulative sum of component variance percentages (Fig. 8).

Figure 7: Percentage of Variance Explained by Principal Components

```
114] # Explained variance confirmation
     print("Variance Explained by all PC's = ", round(sum((pca.explained_variance_ratio_ * 100))))

     Variance Explained by all PC's =  100

115] # Captured variance per PC
     explained_variance = pca.explained_variance_ratio_ * 100

     # Create explained variance df
     df_explained_variance = pd.DataFrame(explained_variance.round(2), columns = ['Percent Variance Explained per PC'], index = ['PC_1', 'PC_2', 'PC_3', 'PC_4', 'PC_5'])

     # View explained variance df
     df_explained_variance
```

	Percent Variance Explained per PC
PC_1	39.86
PC_2	20.48
PC_3	19.95
PC_4	19.57
PC_5	0.13

39.86 + 20.48 = 60.34% Total Variance Explained by PC 1 & PC 2

Figure 8: Heatmap of Percentage of Variance Explained by each Principal Component

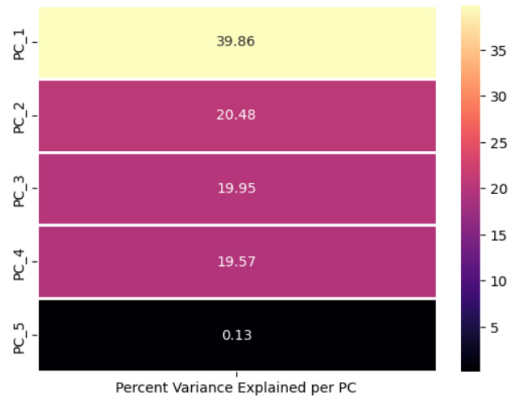
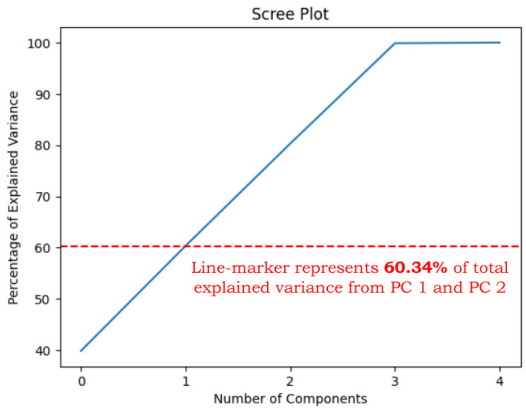


Figure 9: Cummulative Total of Percentage of Explained Variance



D5. Summarize the results of your data analysis.

Principal component analysis was used to perform dimensionality reduction of the WGU Telecom dataset with the goal of identifying principal components to assist with the identification of common characteristics of WGU Telecom subscribers. To complete the analysis, continuous quantitative variables were isolated from the dataset, scaled, and underwent principal component analysis. The results of the PCA were evaluated and significance of components

were determined by the Kaiser criterion, where component features with eigenvalues greater than one were retained.

The PCA examined the following continuous features; Income, Outages (seconds per week), Tenure, Monthly Charges, and Bandwidth (gigabytes per year). The eigenvalues of the PCA are found by computing the explained variances of the five principal components (PC) when the explianed_varainces_ attribute is applied. The Kaiser criterion was employed to determine which principal components should be retained based on the resulting eigenvalues. The first two principal components were kept with qualifying eigenvalues of 1.993 and 1.024 respectively and also captured 60.34% of the total explained variance. Further examination of the principal components one and two reveal statistical significance of explained variance relative to the initial features analyzed. Principal component one is heavily influenced by features Tenure and Bandwidth (gigabytes per year) while principal component two shows positive impact from Income and negative correlations with Outages (seconds per week) and Monthly Charges (Fig. 10).

Figure 10: Principal Component Loadings and Feature Relavance

	PC_1
Income	0.003936
Outage_sec_perweek	0.005858
Tenure	0.705844
MonthlyCharge	0.040748
Bandwidth_GB_Year	0.707159

PC 1 covers 39.86% of total Explained Variance

	PC_2
Income	0.365683
Outage_sec_perweek	-0.682929
Tenure	0.039973
MonthlyCharge	-0.631099
Bandwidth_GB_Year	0.000089

PC 2 covers 20.48% of total Explained Variance

Principal component one accounts for 39.86% of the total explained variance and the statistically significant positive loadings indicate the variables (Tenure, Bandwidth) and the explained variance of principal component one are positively correlated, meaning, an increase in Tenure and Bandwitdth results in an increase of principal component one (Holland, n.d.). Principal component two covered 20.48% of the total explained variance and the mixed positive (Income) and negative loadings (Outages, Monthly Charges) indicate these varaibles are inversely related to principal component two. Both Outages and Monthly Charges carry the most influence relative to the explained variance of PC 2 as represented by their high negatively correlated loadings, -0.683 and -0.631 respectively. Income, on the other hand, has a positively correlated loading of 0.366, but carries less influence on principal component two than Outages and Monthly Charges (values closer to 1 or -1 have a stronger effect on PC). Therefore, relative to principal component two, it is expected to see values Outages and Monthly Charges decrease with an increase in PC 2, while an increase in PC 2 will also see a slight increase in the positively correlated Income variable. These two selected principal components explain the largest amount of variance with reduced input variables and will allow analysts to further extrapolate conclusions, obtain new insights, and target additional business needs.

CODE SOURCES

None

REFERENCES

.
Firmin, S. (2019, July 29). Tidying up with PCA: An Introduction to Principal Components Analysis. Medium. https://towardsdatascience.com/tidying-up-with-pca-an-introduction-to-principal-components-analysis-f876599af383#.

Great Learning Team. (2022, November 21). Understanding Principal Component Analysis and their Applications. Great Learning. https://www.mygreatlearning.com/blog/understanding-principal-component-analysis/.

Holland, S. (n.d.) Principal Component Analysis. Data Analysis in Geosciences. Retrieved August 14, 2023. http://stratigrafia.org/8370/lecturenotes/principalComponents.html#.

Keboola. (n.d.). A Guide to Principal Component Analysis (PCA) for Machine Learning. Keboola. Retrieved August 14, 2023. https://www.keboola.com/blog/pca-machine-learning.

Pierre, S. (2002, August 8). A Step by Step Explanation of Principal Component Analysis (PCA). Builtin. https://builtin.com/data-science/step-step-explanation-principal-component-analysis.

Schork, J. (n.d.). What is Principal Component Analysis (PCA)?. Statistics Globe. Retrieved August 14, 2023. https://statisticsglobe.com/principal-component-analysis-pca.

Scikit-learn. (n.d.). PCA. Scikit-learn 1.3.0 - documentation. Retrieved August 14, 2023. https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html.