D212 – OFM3 TASK 2 – DIMENSIONALITY REDUCTION METHODS

Part I: Research Question

A. Describe the purpose of this data mining report by doing the following:

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

What are the main features of our customers that lead them to leave our company? Since there are so many different features to be taken in consideration here, we are using PCA to reduce our data to the most important characteristics in understanding customer's behavior.

2. Define **one** goal of the data analysis. Ensure that your goal is reasonable within the scope of the scenario and is represented in the available data.

The main goal of our project is to identify which customer's features are relevant to understand customer decision to churn. When stakeholders better understand which features are the most important to prevent churn, they can change marketing and/or customer service strategies to help retaining more customers.

Part II: Method Justification

B. Explain the reasons for using PCA by doing the following:

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

PCA: Principal Component Analysis is a widely used technique to reduce the dimension of your feature space. When we have a system with 10 independent variables, for example, we create 10 "new" variables that are a combination of each of the 10 old ones. We create the new variables in a specific way and order by how well they predict our dependent variable (in our case, Churn). These new 10 variables carry the most valuable parts of our old variables since the new ones are a combination of the old ones. And for that reason, it's not a problem when we drop some of the new created variables. A huge benefit of using PCA is that the new variables are independent of one another[1].

When there are so many features to be looked at in order to understand customer, we apply principal component analysis (PCA) to reduce the amount of features in the problem. Only the most important ones will remain. PCA involves starting with linear algebra operations to better understand the data provided and reducing the number of features that needs to be worked with. Every data needs to be normalized, we need to calculate the eigenvalues and eigenvectors from the correlation matrix. I implemented the Kaiser Criterion to analyze the obtained eigenvalues. All eigenvalues above 1 will be considered good, a projection matrix will be constructed from the selected eigenvectors, transforming the original dataset via the projection matrix to find a new subspace.

2. Summarize **one** assumption of PCA.

One assumption is that we are going to find the most relevant "k" features when k is smaller than the original number of features.

Part III: Data Preparation

C. Perform data preparation for the chosen dataset by doing the following:

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

MonthlyCharge, Bandwidth_GB_Year, Yearly_Equip_Failure, Children, Age, Income, Outage_sec_perweek, Email, Contacts, Tenure.

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

Since PCA seeks to maximize the variance of each component, the first step when implementing PCA is to normalize the data. Our dataset has different kinds of variables with different units. So normalization is a very important first step in this process.

```
# Normalize the data
churn_normalized = (churn_num - churn_num.mean()) / churn_num.std()
pca = PCA(n_components = churn_normalized.shape[1])
churn_numeric = churn_num[['MonthlyCharge', 'Bandwidth_GB_Year',
'Yearly_equip_failure', 'Children', 'Age', 'Income',
                    'Outage_sec_perweek','Email', 'Contacts', 'Tenure']]
pcs_names = []
for i, col in enumerate(churn_numeric.columns):
    pcs_names.append('PC' + str(i + 1))
print(pcs_names)

pca.fit(churn_normalized)
churn_pca = pd.DataFrame(pca.transform(churn_normalized), columns =
pcs_names)

plt.plot(pca.explained_variance_ratio_)
plt.xlabel('Number of Components')
plt.ylabel('Explained Variance')
plt.show()
```

Part IV: Analysis

D. Perform PCA by doing the following:

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

```
#Extract the eigenvalues
cov_matrix = np.dot(churn_normalized.T, churn_normalized) /
churn_num.shape[0]
eigenvalues = [np.dot(eigenvector.T, np.dot(cov_matrix, eigenvector)) for
eigenvector in pca.components_]
```

```
# Plot the eigenvalues
plt.plot(eigenvalues)
plt.xlabel('Number of Components')
plt.ylabel('Eigenvalue')
plt.show()
print(cov_matrix)
```

Covariance Matrix: since I am considering 10 variables, our covariance matrix will be a 10 x 10 matrix.

```
#Extract the eigenvalues
cov_matrix = np.dot(churn_normalized.T, churn_normalized) /
churn_num.shape[0]
eigenvalues = [np.dot(eigenvector.T, np.dot(cov_matrix, eigenvector)) for
eigenvector in pca.components_]
```

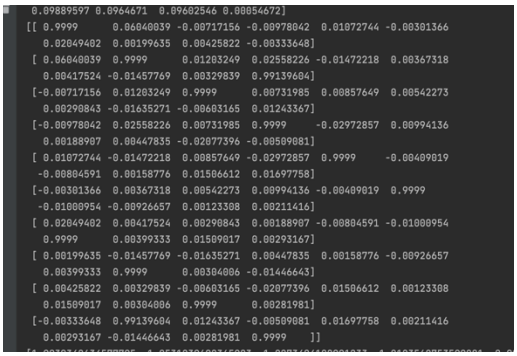

Figure 1: Covariance Matrix

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

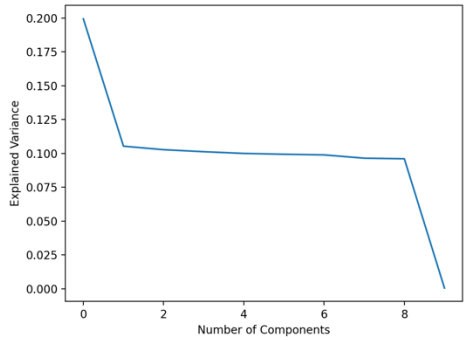

Figure 2: Number of Components vs Explained Variance

```
# Plot the eigenvalues
plt.plot(eigenvalues)
```

```
plt.xlabel('Number of Components')
plt.ylabel('Eigenvalue')
plt.show()
```

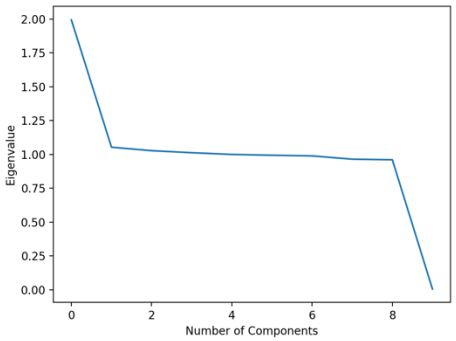

Figure 3: Number of Components vs Eigenvalues

```
print(eigenvalues)
```

Eigenvalues:
[1.9939342634577795, 1.0531239692365983, 1.0273484102991233, 1.0123560753599081, 0.9995955990622569, 0.9935812419893681, 0.988860782514946, 0.9645745267678716, 0.9601585275315967, 0.0054666037805557125]

Eigenvalues are always positive numbers. Therefore, the importance of each value decreases as it approaches 0. An eigenvalue greater than 1, is considered better than average at explaining variance in the dataset. The larger its value, the better the given principal component is at explaining variance[2].

To select the number of principal component to extract from the analysis using the Kaiser Criterion, we must see how many eigenvalues are greater than 1. In this case we have 4 above 1. So, **4** components could be used to reduce the number of original variables and still maintain the maximum level of information from the original data.

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

```
print('Variance of each component is: ', pca.explained_variance_ratio_)
```

Variance: As we have seen in the previous section, we have 4 components:

Variance of each component is: [0.19941337 0.10532293 0.10274512 0.10124573 0.09996956 0.09936806 0.09889597 0.0964671 0.09602546 0.00054672]

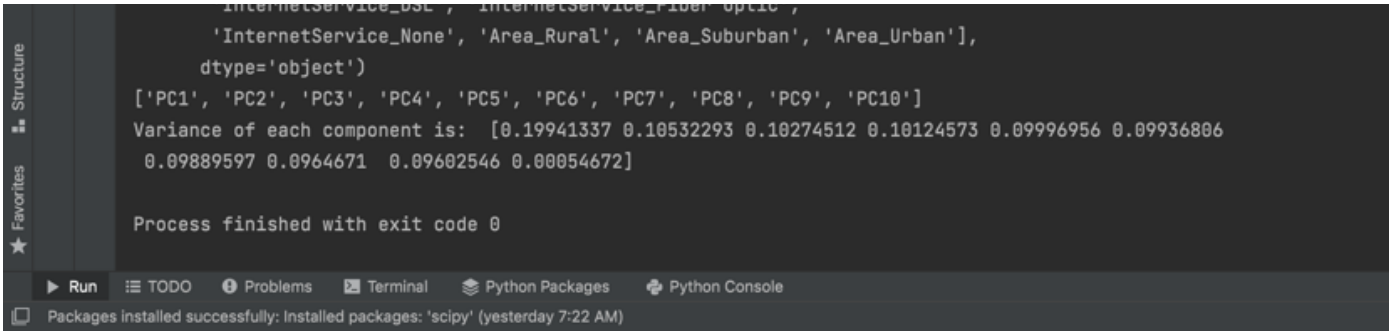

Figure 4: Variance of Each of the Principal Components

PC1 Variance: 0.19941337 (PC1V)
PC2 Variance: 0.10532293 (PC2V)
PC3 Variance: 0.10274512 (PC3V)
PC4 Variance: 0.10124573 (PC4V)

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

Total Variance:

The total variance is **the sum of variances of all individual principal components**[1].

0.19941337 + 0.10532293 + 0.10274512 + 0.10124573 = **0.5087 (TV)** It means that 50.87% of all variance is explained by the 4 components.

5. Summarize the results of your data analysis.

4 components could be used to reduce the number of original variables and still maintain the maximum level of information from the original data. We have to analyze 4 variables instead of 10!

Steps performed to achieve the conclusion above: first a few continuous variables were selected, then they were normalized, a covariance matrix was calculated, we made plots to understand the explained variance and eigenvalues, using the Kaiser Criterion we kept only 4 components from our PCA analysis. These components can be used in further data mining analysis in order for the stakeholders to have a better understanding of how the variables can be related to the churn problem.

```
#Select the fewest components
for pc, var in zip(pcs_names, np.cumsum(pca.explained_variance_ratio_)):
    print(pc, var)
```

Figure 5: Principal Components Variance

From the rotation matrix we see that MonthlyCharge, Bandwidth and Tenure are the most important variables in understanding churn.

```
#Creating Rotation
rotation = pd.DataFrame(pca.components_.T, columns = pcs_names, index =
churn_numeric.columns)
print('Rotation Matrix is: ', rotation)
```

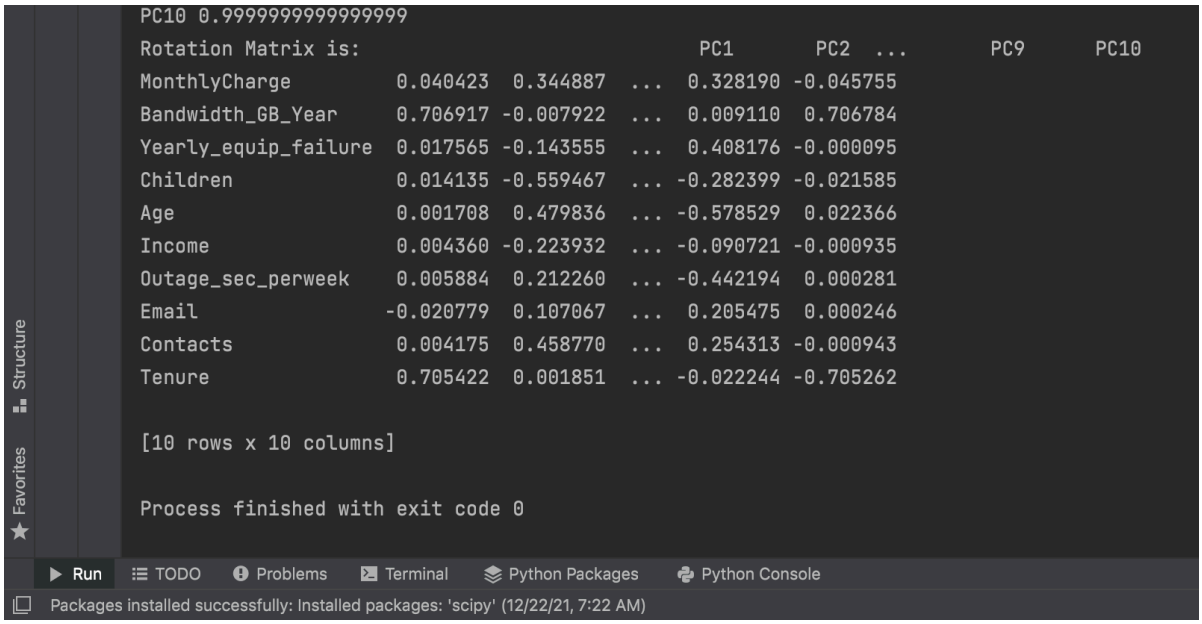

Figure 6: Rotation Matrix

Part V: Attachments

E. Record the web sources used to acquire data or segments of third-party code to support the analysis. Ensure the web sources are reliable.

I used some of the same code from my previous report for D206.

2021, August 30th BISCHOFF, Bianca Assessment Document for D206 - Bianca Bischoff

F. Acknowledge sources, using in-text citations and references, for content that is quoted, paraphrased, or summarized.

H. Sources

[1] Brems, Matt. (2017, April 17) A One-Stop Shop for Principal Component Analysis https://towardsdatascience.com/a-one-stop-shop-for-principal-component-analysis 5582fb7e0a9c

[2] Larose, C. D. & Larose, D. T. (2019). Data Science: Using Python and R. John Wiley & Sons, Inc.

[3] (2017, Dec 11th) CHEPLYAKA, Roman, Explained Variance in PCA, https://ro-che.info/articles/2017-12-11-pca-explained-variance