Dimension Reduction of Telcom Customer Churn Data

Mike Mattinson

Western Governors University

D212: Data Mining II

Task 2: Dimension Reduction

Dr. Kesselly Kamara

May 2, 2022

Abstract

Telecom customer data has 50 attributes defining each customer, this analysis will use dimension reduction techniques to identify the most influential variables. Data source: Wgu.edu Telecom Churn data (N: 10,000).

Keywords: Telecom. Churn. Data Mining. Dimension Reduction. PCA.

Dimension Reduction of Telcom Customer Churn Data

Scenario 1

One of the most critical factors in customer relationship management that directly affects a company's long-term profitability is understanding its customers. When a company can better understand its customer characteristics, it is better able to target products and marketing campaigns for customers, resulting in better profits for the company in the long term.

You are an analyst for a telecommunications company that wants to better understand the characteristics of its customers. You have been asked to use principal component analysis (PCA) to analyze customer data to identify the principal variables of your customers, ultimately allowing better business and strategic decision-making.

Part I: Research Question

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

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

What are the most influential features of the Telecom customer data?

A2. 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.

Use principal component analysis (PCA) to identify the most influential features of the customer data. The primary dataset consists of 10,000 customer records with 50 attributes each. The overall steps to the analysis are below:

PCA Pseudo Code:

1. R <= import .csv file
2. S <= Drop unwanted data from R
3. target <= 'Churn'
4. Y <= S.loc[:, S.columns == target]
5. X <= S.loc[:, S.columns != target]
6. D <= create dummy variables from X
7. Z <= standardize D
8. A<= remove highly correlated features
9. Create covariance matrix
10. Calculate Eigenvalues
11. Sort Eigenvalues

Part II: Technique Justification

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

B1. Explain how the PCA analyzes the selected the selected dataset. Include expected outcomes.

From the t

Here is the code used to create Figure 1:

```
# create scatter plot of lost customer data
fig, ax = plt.subplots(figsize =(7, 5))
plt.plot(df["TEN"], df["MCH"], marker="x",
linestyle="")
plt.xlabel("Tenure")
plt.ylabel("Monthly Charge")
plt.title("Lost Customers (Churn='Yes')")
fig.savefig("figures/fig_1", dpi=150)
```

B2. Summarize one assumption of PCA.

Data scie

Part III: Data Preparation

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

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

Using data preparation and exploratory data analysis, the following list of variables were determined to be relevant to the analysis. Using a helper function, these numerical variables are described showing whether it is continuous or categorical data:

```
# describe variables as continuous or categorical
describe_dataframe_type(df_numerical)

1. INC is numerical (CONTINUOUS) - type: float64.
  Min: 348.670  Max: 189938.400  Std: 28623.988

2. OUT is numerical (CONTINUOUS) - type: float64.
  Min: 0.232  Max: 21.207  Std: 2.970

3. TEN is numerical (CONTINUOUS) - type: float64.
  Min: 1.000  Max: 71.646  Std: 15.577

4. MCH is numerical (CONTINUOUS) - type: float64.
  Min: 92.455  Max: 290.160  Std: 41.268

5. BAN is numerical (CONTINUOUS) - type: float64.
  Min: 248.179  Max: 7096.495  Std: 1375.370
```

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

The cleaned dataset is saved to an external text file. Table 1 is a list of the file showing the first 10 rows:

Table 1

Cleaned Dataset (First 10 rows)

Source: cleaned.csv

Step 7. Find highly correlated variables using a correlation matrix

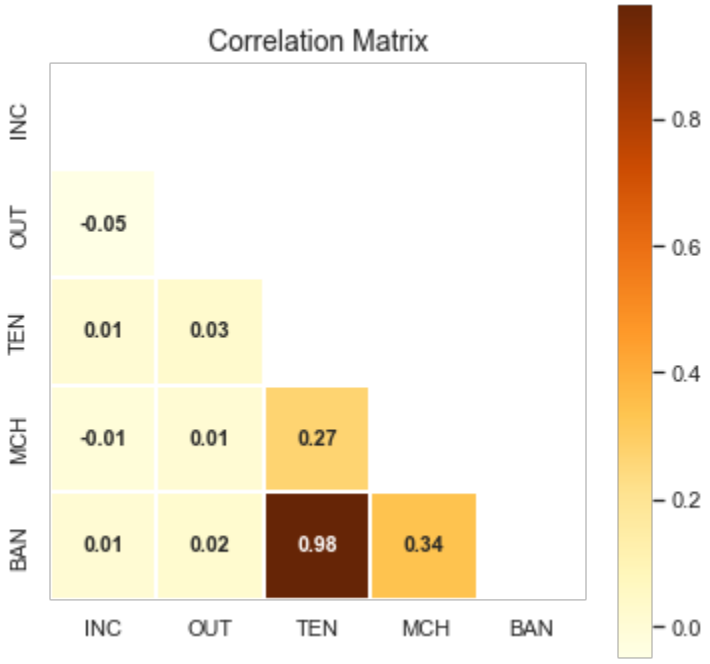

Figure 1 Correlation Matrix

Here is the code to generate the correlation matrix:

```
# use heatmap graph to identify highly correlated variables
def Generate_heatmap_graph(corr, chart_title,
mask_uppertri=False ):
    """ Based on features , generate correlation matrix """
    mask = np.zeros_like(corr)
    mask[np.triu_indices_from(mask)] = mask_uppertri
    fig,ax = plt.subplots(figsize=(6,6))
    sns.heatmap(corr
                , mask = mask
                , square = True
                , annot = True
                , annot_kws={'size': 10.5, 'weight' : 'bold'}
                , cmap=plt.get_cmap("YlOrBr")
                , linewidths=.1)
    plt.title(chart_title, fontsize=14)
    plt.show()

Generate_heatmap_graph(
    round(df_numerical.corr(),2),
    chart_title = 'Correlation Matrix',
    mask_uppertri = True)
```

Step 9. Standardize remaining numerical data

```
# standardize remaining numerical data
scaler = StandardScaler()
scaled_features = scaler.fit_transform(df_final.values)
df_standardized = pd.DataFrame(scaled_features,
        index=df_final.index,
      columns=df_final.columns)
df_standardized.describe().round(2)
```

STD	INC	OUT	TEN	MCH
count	2650.00	2650.00	2650.00	2650.00
mean	-0.00	0.00	-0.00	-0.00
std	1.00	1.00	1.00	1.00
min	-1.39	-3.29	-0.78	-2.59
25%	-0.73	-0.67	-0.58	-0.77
50%	-0.23	-0.01	-0.34	0.02
75%	0.49	0.66	0.04	0.81
max	5.24	3.77	3.76	2.20

Here is the code to create the boxplot:

```
# use boxplot to look for outliers
fig, ax = plt.subplots(figsize =(7, 5))
ax = df_standardized.boxplot(vert=False)
```

Part IV: Analysis

D. Perform the data analysis and report on the results by doing the following:

D1. Describe the analysis technique you used to appropriately analyze the data. Include screenshots of the intermediate calculations you performed.

The following steps were used to appropriately analyze the data using k-means clustering:

D2. Provide the code used to perform the clustering analysis technique from part 2.

Here is the adapted code used to create the Knee Plot (Arvai, 2022):

```
# create knee plot, adapted code (Arvai, 2022)
kmeans_kwargs = {
    "init": "random",
    "n_init": 10,
    "max_iter": 300,
    "random_state": 42 }
sse = [] # list of SSE values for each k
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, **kmeans_kwargs)
    kmeans.fit(scaled_features)
    sse.append(kmeans.inertia_)
fig, ax = plt.subplots(figsize =(7, 5))
knee = KneeLocator(range(1, 11), sse, curve="convex",
direction="decreasing")
plt.plot(range(1, 11), sse)
plt.xticks(range(1, 11))
plt.xlabel("Number of Clusters")
plt.ylabel("SSE")
plt.title("Knee Plot")
plt.axvline(x=kl.elbow, color='green', ls=':', lw=2,)
fig.savefig("figures/fig_2", dpi=150)
```

Here is the adapted code to show the optimum point on knee plot (Arvai, 2022):

```
# optimum point on knee plot
'Optimum: ({}, {:.3f})'.format(knee.elbow, sse[knee.elbow-1])

Out[]: 'Optimum: (4, 5326.264)'
```

Here is the adapted code used to generate the final cluster plot (Arvai, 2022):

```
# final K-means analysis plot
fig, ax = plt.subplots(figsize =(7, 5))
title = 'K-Means Clustering (k=' + str(n_clusters) + ') for
Lost Customers'
ax.scatter(x=df_standardized['TEN'],y=df_standardized['MCH'],
    c=kmeans.labels_,cmap='brg')
ax.scatter(x=kmeans.cluster_centers_[:,2],
    y=kmeans.cluster_centers_[:,3],
    color='black', marker='X',s=400 )
ax.set_xlabel('Tenure (standardized)')
ax.set_ylabel('Monthly Charge (standardized)')
plt.title(title)
fig.savefig("figures/fig_3", dpi=150)
```

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. (see References below)

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

G. Demonstrate professional communication in the content and presentation of your submission.

References

Albon, C. (2017). Drop Highly Correlated Features. Retrieved from https://chrisalbon.com/code/machine_learning/feature_selection/drop_highly_correlated_features/

Alkhayrat, M., Aljnidi, M., & Aljoumaa, K. (2020). A comparative dimensionality reduction study in telecom customer segmentation using deep learning and PCA. Retrieved from https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-0286-0

Arvai, K. (2022). K-Means Clustering in Python: A Practical Guide. Retrieved from https://realpython.com/k-means-clustering-python/#: :text=The SSE is defined as,try to minimize this value.&text=The purpose of this figure,centroids is an important step.

Brownlee, J. (2019). How to Calculate Principal Component Analysis (PCA) from Scratch in Python. Retrieved from https://machinelearningmastery.com/calculate-principal-component-analysis-scratch-python/

Bruce, P. C., Gedeck, P., Shmueli, G., & Patel, N. R. (2019). *Data Mining for Business Analytics Concepts, Techniques and Applications in Python.* Wiley & Sons, Incorporated, John.

Bruce, P., Bruce, A., & Gedeck, P. (2020). *Practical Statistics for Data Scientists 50+ Essential Concepts Using R and Python.* O'Reilly Media, Incorporated.

Fenner, M. (2018). *Machine Learning with Python for Everyone.* Addison Wesley.

Galarnyk, M. (2017). PCA using Python (scikit-learn). Retrieved from https://towardsdatascience.com/pca-using-python-scikit-learn-e653f8989e60

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems.* O'Reilly Media.

h1ros. (2019). Drop Highly Correlated Features. Retrieved from https://h1ros.github.io/posts/drop-highly-correlated-features/

Larose, C. D., & Larose, D. T. (2019). *Data Science.* Wiley.

Parveez, S., & Iriondo, R. (2018). Principal Component Analysis (PCA) with Python Examples — Tutorial. Retrieved from https://pub.towardsai.net/principal-component-analysis-pca-with-python-examples-tutorial-67a917bae9aa

ProjectPro. (2022). How to drop out highly correlated features in Python? Retrieved from https://www.projectpro.io/recipes/drop-out-highly-correlated-features-in-python

Sawlani, D. (2017). Customer Churn Prediction Model. Retrieved from https://rstudio-pubs-static.s3.amazonaws.com/344958_a3c32d38a83043c8ae8fc13cd6abbda4.html

Sharma, A. (2021). PCA or Principal Component Analysis on Customer Churn Data. Retrieved from https://medium.com/data-science-on-customer-churn-data/pca-or-principal-component-analysis-on-customer-churn-data-d18ca60397ed

Srinivas. (2022). Reducing Telecom Churn with PCA and Modeling. Retrieved from https://www.kaggle.com/code/manoharsrinivas/reducing-telecom-churn-with-pca-and-modeling/notebook

VanderPlas, J. (2016). *In Depth: Principal Component Analysis.* O'Reilly. Retrieved from https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysis.html