Cali Ciccarelli

D206 Data Cleaning

11/29/2023

Dr. Straw

Data Cleaning Task Project

Part 1: Research Question

1. What are the factors that may influence the churn rate of the company? The company wants to keep as many customers as possible to maximize sales, so they want to look at some basic demographics and statistics of the customers that did churn and did not.
2. The data types for each variable are as follows:   
   (Straw, 2023)

CaseOrder: Qualitative/Categorical, The order of the original raw data file. Ex. 6

Customer\_id: Qualitative/Categorical, Used to identify each customer. All unique. Ex. K409198

Interaction: Qualitative/Categorical, Used to identify services with a customer. Unique. Ex. aa90260b-4141-4a24-8e36-b04ce1f4f77b

City: Qualitative/Categorical, Address found for customer, ex. Oakland

State: Qualitative/Categorical, Address found for customer, ex. OH

County: Qualitative/Categorical, Address found for custome, ex. Vilas

Zip: Qualitative/Categorical, Address found for customer, ex. 601162

Lat: Quantitative/Numerical, Continuous, coordinates of customers location, ex. 39.96636

Lng: Quantitative/Numerical, Continuous, coordinates of customers location, ex. -81.30311

Population: Quantitative/Numerical, Discrete, population of people within a mile radius of customer address, ex. 2093

Area : Qualitative/Categorical, Nominal, description of the customers area, ex. suburban

Timezone: Qualitative/Categorical, Nominal, Customer’s timezone, ex. America/Puerto\_Rico

Job : Qualitative/Categorical, Nominal, Customer’s job title, ex. Engineer

Children: Quantitative/Numerical, Discrete, number of kids customer has, ex. 2

Age: Quantitative/Numerical, Discrete, age of customer reported, ex. 52

Education: Qualitative/Categorical, Nominal, furthest education received by customer, ex. Bachelor’s Degree

Employment: Qualitative/Categorical, Nominal, whether or not customer is currently employed

ex. Retired

Income: Quantitative/Numerical, Continuous, The income a customer brings in each year, ex. 50231.4

Marital: Qualitative/Categorical, Nominal, relationship status, ex. Widowed

Gender: Qualitative/Categorical, Nominal, gender identity of customer, ex. Female

Churn: Qualitative/Categorical, Nominal, whether or not customer ended their services within the last month, ex. yes

Outage\_sec\_perweek: Quantitative/Numerical, Continuous, avg seconds a customer experienced a power outtage by week, ex. 11.83511269

Email: Quantitative/Numerical, Discrete, number of emails customer received from the company within the year. Ex. 2

Contacts: Quantitative/Numerical, Discrete, amount of times customer reached out for tech support, ex. 6

Yearly\_equip\_failure: Quantitative/Numerical, Discrete, amount of times in a year that a customer had equipment that had to be replaced or reset up after experiencing failure, 3

Techie: Qualitative/Categorical, Nominal, If a customer believes they have technical experience/knowledge, ex. yes

Contract: Qualitative/Categorical, Nominal, The type of length contract a customer has, ex. Month-to-month

Port\_modem: Qualitative/Categorical, Nominal If a customer has a portal modem or not, ex. yes

Tablet: Qualitative/Categorical, Nominal, If the customer owns a tablet like an Ipad, ex. yes

InternetService: Qualitative/Categorical, Nominal, The ISP type of the customer ex. Fiber optic

Phone: Qualitative/Categorical, Nominal, if the customer has phone service, ex. no

Multiple: Qualitative/Categorical, Nominal, Does a customer have multiply lines for service, ex. no

OnlineSecurity: Qualitative/Categorical, Nominal, If the customer has a security add-onn, ex. no

OnlineBackup : Qualitative/Categorical, Nominal, If the customer has a back-up add on ex. no

DeviceProtection : Qualitative/Categorical, Nominal, If the customer pays for device protection, ex. yes

TechSupport: Qualitative/Categorical, Nominal, if the customer pays for the add on for tech support, ex no

StreamingTV: Qualitative/Categorical, Nominal, If customer has streaming tv, ex. yes

StreamingMovies: Qualitative/Categorical, Nominal, If customer has streaming movies, ex. yes

PaperlessBilling: Qualitative/Categorical, Nominal If customer uses paperless billing, ex. yes

PaymentMethod: Qualitative/Categorical, Nominal, what is the customer’s payment method type, ex. Credit card(automatic)

Tenure: Quantitative/Numerical, Continuous, how long the customer has stayed with provider in the number of months ex. 17.08722662

MonthlyCharge: Quantitative/Numerical, Continuous, How much a customer is charged per month on

average ex. 154.017102

Bandwidth\_GB\_Year: Quantitative/Numerical, Continuous, GB of data used on average per year, ex.

904.5361102

item1 : Qualitative/Categorical, Ordinal, survey question on the importance of a timely response.

1 being most important and 8 being the least. Ex. 8

Item2: Qualitative/Categorical, Ordinal, survey question on the importance of a timely fix.

1 being most important and 8 being the least. Ex. 7

Item3: Qualitative/Categorical, Ordinal, survey question on the importance of a timely replacement.

1 being most important and 8 being the least. Ex. 6

Item4: Qualitative/Categorical, Ordinal, survey question on the importance of reliability.

1 being most important and 8 being the least. Ex. 5

Item5: Qualitative/Categorical, Ordinal, survey question on the importance of options.

1 being most important and 8 being the least. Ex. 4

Item6: Qualitative/Categorical, Ordinal, survey question on the importance of a respectful response.

1 being most important and 8 being the least.ex. 3

Item7: Qualitative/Categorical, Ordinal, survey question on the importance of a courteous

exchange. 1 being most important and 8 being the least. Ex. 2

Item8: Qualitative/Categorical, Ordinal, survey question on the importance of evidence of

active listening. 1 being most important and 8 being the least. Ex. 1

Part II: Data-Cleaning Plan

1. 1. When cleaning my data I will first ensure there is no duplicated data. For this specific research question, we are looking at each customer so we will want to remove any records where the Customer\_id’s are duplicated. I will use the command “.duplicated” to locate the duplicated data. I will then assess the quality of the data and look for null values and outliers for numerical data. For the null values, I will use the command “.isna” and “.sum” to locate the null values in the dataset. For outliers, I will use the “stats.zscore” function. I will also use the “.abs” value command. Then I will look for values that had a z\_score greater than 3. I also changed the z score to 4 and then 5 so I could review the highest/lowest values for each column an ensure they were within reason. I will also check the categorical variables and ensure there are no typos/errors with the data by reviewing the unique values for each. I will do this by using the “.unique” command.

2. For duplicates, I examined the Customer\_id because this will identify if any customers were put into the data set twice. For the numerical values, it is vital to look for nulls and outliers. These values can inaccurately impact analysis so they need to be identified and examined. Calculating the z-scores helps to find outliers with ease. I also thought it would be important to look at the unique values for each categorical variables because this will help identify if there are any typos or errors that occurred. For example, Item1-8 should only have number 1-8. If there were any numbers outside of this, they would be inaccurate. It could also be possible that caps/typos occurred with categories like marital status. Any typos could cause there to be more values than needed and would make analysis difficult and unclear.

3. I have chosen to use Python with Pandas and I am using the Jupyter Notebook workspace to work on the data cleaning. I have chosen python because it is becoming increasingly popular and is more versatile. According to statistica, it is the third most used programming language. Meanwhile, R ranks 21st (Vailshery, 2023). While R does have some striking benefits with its data analytic abilities, Python is more valuable to learn today. I found the Jupyter Notebook to be the best option for me. I originally began with the program “Spyder”, but I found that my code would often not run through even when it was correct. The Jupyter notebook has not had any problems. It was helpful to have the bar on the left lights up orange when the code needs to be fixed. I also used pandas to easily bring in the database. I imported numpy so that I could calculate the z score using the absolute value. I also imported stats from scipy so that I could calculate the z-score. Finally, for PCA, I imported PCA from sklearn.decomposition,

imported matplotlib.pyplot and

imported seaborn.

4. import pandas as pd

import numpy as np

from scipy import stats

from sklearn.decomposition import PCA

import matplotlib.pyplot as plt

import seaborn as sns

df = pd.read\_csv("/Users/cali/Downloads/d206-churn/churn\_raw\_data.csv")

df.shape

df['index']=pd.Series(range(0,10000)) #added index starting at 0 since python starts at the 0 place instead of 1

dupcustomer= df[df.duplicated(['Customer\_id'])] #checking for duplicated records

print(dupcustomer)

print(df.isna().sum()) #checking for NaN values.

z\_population = np.abs(stats.zscore(df['Population'])) #(Cloud, 2023)

outliers\_population = df[(z\_population > 3)]

print(outliers\_population[['Population','Area', 'City']])

z\_children = np.abs(stats.zscore(df['Children']))

outliers\_children = df[(z\_children > 3)]

print(outliers\_children[['Customer\_id','Children']])

filter\_children = df[df['Children'] > 12]

select\_children = filter\_children[['Age', 'Children']]

print(select\_children) #shows how many customers have more than 12 children. None found. Outliers are being left untouched.

z\_age = np.abs(stats.zscore(df['Age']))

outliers\_age = df[(z\_age > 3)]

print(outliers\_age[['Customer\_id','Age']])

filter\_age = df[df['Age'] < 18]

select\_age = filter\_age[['Customer\_id','Age']]

print("Under 18 Years Old:")

print(filter\_age)

z\_income = np.abs(stats.zscore(df['Income']))

outliers\_income = df[(z\_income > 3)]

print(outliers\_income[['Job','Income']])

filter\_income = df[df['Income'] > 200000.00] #only 3 records return and with job seem reasonable. Leaving untouched.

select\_income = filter\_income[['Job', 'Income']]

print(select\_income)

z\_tenure = np.abs(stats.zscore(df['Tenure']))

outliers\_tenure = df[(z\_tenure > 3)]

print(outliers\_tenure) #no outliers found

z\_bandwidth = np.abs(stats.zscore(df['Bandwidth\_GB\_Year']))

outliers\_bandwidth = df[z\_bandwidth > 3]

print(outliers\_bandwidth)

z\_outage = np.abs(stats.zscore(df['Outage\_sec\_perweek']))

outlier\_outage = df[z\_outage > 5]

print(outlier\_outage[['Outage\_sec\_perweek', 'Area']])

z\_email = np.abs(stats.zscore(df['Email']))

outlier\_email = df[z\_email > 3]

print(outlier\_email[['Email', 'Customer\_id']])

z\_contacts = np.abs(stats.zscore(df['Contacts']))

outlier\_contacts = df[z\_contacts > 5]

print(outlier\_contacts[['Customer\_id', 'Contacts']])

z\_eqfail = np.abs(stats.zscore(df['Yearly\_equip\_failure']))

outlier\_eqfail = df[z\_eqfail > 5]

print(outlier\_eqfail[['Customer\_id', 'Yearly\_equip\_failure']])

z\_monthlycharge= np.abs(stats.zscore(df['MonthlyCharge']))

outlier\_monthlycharge =df[z\_monthlycharge > 3]

print(outlier\_monthlycharge[['Customer\_id', 'MonthlyCharge']])

print(df['Area'].unique())

print(df['Timezone'].unique())

print(df['Job'].unique())

print(df['Education'].unique())

print(df['Employment'].unique())

print(df['Marital'].unique())

print(df['Gender'].unique())

print(df['Churn'].unique())

print(df['Techie'].unique())

print(df['Contract'].unique())

print(df['Port\_modem'].unique())

print(df['Tablet'].unique())

print(df['InternetService'].unique())

print(df['Phone'].unique())

print(df['Multiple'].unique())

print(df['OnlineSecurity'].unique())

print(df['OnlineBackup'].unique())

print(df['DeviceProtection'].unique())

print(df['TechSupport'].unique())

print(df['StreamingTV'].unique())

print(df['StreamingMovies'].unique())

print(df['PaperlessBilling'].unique())

print(df['PaymentMethod'].unique())

print(df['item1'].unique())

print(df['item2'].unique())

print(df['item3'].unique())

print(df['item4'].unique())

print(df['item5'].unique())

print(df['item6'].unique())

print(df['item7'].unique())

print(df['item8'].unique())

See code attached

**Part III: Data Cleaning**

D.

1. When looking for duplicated records, it was found that there were no duplicated customers in the dataset. When looking for NaN/null values, there were many found. There were 2495 null values for Children, 2475 null values for Age, 2490 null values for Income, 2477 null values for Techie, 1026 null values for phone, 991 null values for TechSupport, 931 null values for Tenure, and 1021 null values for Bandwidth\_GB\_Year. There were also outliers found using the Z-score method. There were outliers found for Population, Children, Income, Email, Outtage\_sec\_perweek, Contacts, Yearly\_equip\_failure, and MonthlyCharge. There were many outliers found and I had to take a deeper look at each column and examine the values to decide if the outliers were true or not. I filtered the “Children” column to print a filtered selection of customers who had more than 12 children. The dataset came back empty. For income there were many outliers as well. I filtered income to show any records of customers who made more than $200,000 per year and I selected the Job and Income to show. I found that there were only 3 customers who made more than $200,000. None of which made more than $260,000 and the job titles seemed within reason for the incomes. For Outage\_sec\_perweek, there were many outliers as well. Instead of filtering, this time I edited the z\_score filter for the outliers to be 5. (Please note, I used the absolute value when calculating z-scores so I do not need to check negative values). This showed me that the highest values did not pass 50 seconds. I found this to reasonable as some areas tend to get more outages than others. There were only around 12 outliers for emails, although again, they were within reason. The data dictionary states the emails are for correspondence or marketing. It could be very possible that a customer is new, so they only had 1-2 emails. It is also possible that a customer had a lot of issues or troubleshooting in which they received 22-23 emails. There were many outliers again with contacts and yearly\_equip\_failure. I changed the z-score filter from 3 to 5 and was able to examine the top/low values. Contacts did not exceed 7 and Yearly\_equip\_failure did not exceed 6. Finally, there were only 3 outliers for monthly charge. The highest was $315.878600.
2. Since there were no duplicated values, no action was taken. If I had found any, I would have used the .drop command to remove the duplicated records. For the N/A values for “Income” and“Age”, I replaced the N/As with the mean for that given column. I chose the mean because I decided it would be too much data to drop out of the dataset. Especially when there are so many other variables for each record. I also used the mean because it will not affect the normal distribution too much. For, the N/A values for “Children”, I replaced them with the median instead of the mean. This is because customers cannot have a part of a child (ex. 2.015). There is also not an even distribution for the children, so it is better to use the median rather than the mean. For the categorical variables, I noticed they were all binary, yes/no questions. The “Techie” column refers to whether or not a customer believes they have technical knowledge. I replaced the n/a values with the mode (most replied answer, “No”). For “Phone” and “TechSupport”, I decided it was likely that N/A, or leaving this value blank, likely meant that they do not have the service at all. Therefore, I decided to fill the N/A values for “Phone” and “TechSupport” with “No”. Next, I moved onto the outliers. With population, there was an even spread of values that went up to around 100,000 people. There were no value that appeared untrue such as 0 or 700,000, so I made the choice to leave the outliers in place. I decided to mark the children outliers as true values because it is very possible to have 8-12 children. It is uncommon, but not necessarily an untrue value. I also found the income outliers to be within reason, so I left them in place. After examination of Outage\_per\_week, I decided to leave the values. Nothing stood out as untrue. I, again, left alone the outliers for Email, Yearly\_epuip\_failure, and MonthlyCharge. Upon examining the furthest outliers, they all appeared possible and understandable. True outliers are great for analysis and can show some vital trends. When checking for typos or any messy looking values for categorical columns, all appeared clean. I would recommend to stakeholders that we create a new column for job field and break it down into around 10 categories like “healthcare”, “law”, “education”, etc. The way the job values are expressed, there are a large amount of unique values which will be difficult to visualize and analyze.
3. I did not have any action required for duplicates, I replaced all NaN values. For “Children”, the N/As have been replaced with the median. For “Income” and “Age”, the N/As have been replaced with the mean values. For the “Techie” column, the N/As have been replaced with the mode value. Finally, “Phone” and “TechSupport” N/As have been filled with “No”. I left the outliers in place. Now that the data is cleaned, there are no more null values for any columns. Beyond the null values, the data was not altered much. A large part of cleaning data is checking it. There are not always going to be data that needs to be treated. The data is now verified to be ready for analysis.
4. Treatment code:

import pandas as pd

import numpy as np

from scipy import stats

from sklearn.decomposition import PCA

import matplotlib.pyplot as plt

import seaborn as sns

df = pd.read\_csv("/Users/cali/Downloads/d206-churn/churn\_raw\_data.csv")

median\_children= df ['Children'].median()

print(median\_children)

df['Children'] = df['Children'].fillna(median\_children)

mean\_age= df['Age'].mean()

print(mean\_age)

df['Age'] = df['Age'].fillna(mean\_age)

mean\_income = df['Income'].mean()

print(mean\_income)

df['Income'] = df['Income'].fillna(mean\_income)

mean\_tenure= df['Tenure'].mean()

print(mean\_tenure)

df['Tenure'] = df['Tenure'].fillna(mean\_tenure)

mean\_bw = df['Bandwidth\_GB\_Year'].mean()

print(mean\_bw)

df['Bandwidth\_GB\_Year'] = df['Bandwidth\_GB\_Year'].fillna(mean\_bw)

df['Techie'] = df['Techie'].fillna('No')

df['Phone']= df['Phone'].fillna('No')

df['TechSupport'] = df['TechSupport'].fillna('No')

See attached for treatment code

1. See attached files for CSV of cleaned data
2. The disadvantage of using the Customer\_id to search for duplicated records could be that there was the same customer but with a different customer\_id in place. It is possible that I could have missed a potential repeated record. For the N/A values, there a certainly some disadvantages to using the mean/median values. There were a large number of N/A values. It could have been true that N/A for children or income meant 0. I decided to use the median/mean because there were already many who placed 0 for children or income. However, it is still possible that N/A meant 0 for some. Filling with mean/median values may end up centralizing the data set too much and may have caused there to be more “outliers” than there were. Leaving all outliers in place may have caused data to be skewed in an incorrect direction. I was unable to confirm or deny any outliers or discuss N/A values due to not having communication with the data collector(s). The most accurate course would have been to reach out to the customers and fill the n/a values with correct numbers and confirm the outliers. This was not possible in this case and thus, limitations to the dataset are known.
3. Filling the N/A values can cause the analysis to be inaccurate in several ways. For example, if they were trying to find the age that had the most churn rate, they may automatically be given the mean age, rather than the true answer. If the the N/As for the services “Phone” and “TechSupport” were actually not “No” values, this could also lead to some incorrect assumptions about the effect of the services on the churn rate. Leaving the outliers in place may also affect analysis. For example, if they were to find an individual with the highest number of equipment failures did not discontinue service when the Yearly\_equip\_failure was an untrue outlier, this will show inaccuracies. They may end up determining that those with high equipment failure stay with the company more when that was not the case at all.

E1:

I completed the PCA with the 7 numerical, continuous variables of the churn data set. This included 'Lat','Lng','Income','Outage\_sec\_perweek','Tenure','MonthlyCharge','Bandwidth\_GB\_Year'.

Below is the loadings.

A screenshot of a computer

Description automatically generated

2.

The principal components that are the most important are PC1, PC2, PC3, and PC4. I determined this by using the eigenvalues. The first 4 components were found to have a value greater than 1. See below for the scree plot. I chose this method because it is straightforward to read the plot despite the math behind the computation being complex.

A graph with a line

Description automatically generated

3. PCA is helpful because it helps us determine what components will be helpful to look at and what relationships are not meaningful. This can simplify the amount of data we are looking at and can decrease the time a cost it will take to complete an analysis. With this project PC4-PC7 are unimportant relations to focus on for the data analysis.

F. Panopto video

https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=33acc25f-ff73-430a-bb3a-b0c7015fd4f2

G. Sources for Code

Cloud, S. (2023). Retrieved from https://saturncloud.io/blog/how-to-detect-and-exclude-outliers-in-a-pandas-dataframe/

Middleton, K. (n.d.)  *D206 - Getting Started with D206 | PCA*. Retrieved from <https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=3bcc452f-fa35-43be-b69f-b05901356f95>

H. Sources for Content

Straw, Dr. E. (2023). Retrieved from https://westerngovernorsuniversity.sharepoint.com/sites/DataScienceTeam/Shared%20Documents/Forms/AllItems.aspx?id=%2Fsites%2FDataScienceTeam%2FShared%20Documents%2FGraduate%20Team%2FD206%2FStudent%20Facing%20Resources%2FDr%20Straw%20breakdown%20of%20data%20types%2Epdf&parent=%2Fsites%2FDataScienceTeam%2FShared%20Documents%2FGraduate%20Team%2FD206%2FStudent%20Facing%20Resources

Vailshery, L. S. (2023). Retrieved from https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/