D206 Performance Assessment

Shanay Murdock

smurd32@wgu.edu

Student ID: 011377935

D206 – Data Cleaning

Western Governors University

July 12, 2024

A. Research Question

For this project, I have opted to continue using the churn_raw_data.csv dataset out of intrigue, as I started working with the dataset in D205. I also used to work in the data sector for a telecommunications company. I have obtained a new copy of the raw dataset for D206.

As it is more profitable to retain existing customers than to gain new customers (keeping in mind that adding new customers is still critical to a company's growth), the research focus for this project is to understand what factors contribute to customer churn. If these factors can be identified, it might be possible to recommend marketing and service improvements to decrease the number of customers that leave the service provider. This can help C-level executives decide what products and services to invest in, when, and to what degree. These decisions allow the company to evolve, meet its clientele's needs, and remain competitive in a flooded market.

B. Required Variables

The following section describes the 51 variables in churn_raw_data.csv file, including the column header, the type of data corresponding to each column header, a description of the data in the column, and an example of the data pulled directly from the CSV file.

Note: The UID variable listed on the Data dictionary is not present in the CSV file upon download.

Column Header	Data Type	Description	Example Data
CaseOrder	Quantitative	An index to preserve the original order of the raw data	1
Customer_id	Qualitative	A unique customer identifier	K409198
Interaction	Qualitative	A unique identifier related to customer transactions, technical support, and service sign-ups	aa90260b-4141-4a24-8e36-b04ce1f4f77b
City	Qualitative	City of customer's residence per the billing statement	Point Baker
State	Qualitative / Nominal	State of customer's residence per the billing statement	AK
County	Qualitative	County of customer's residence per the billing statement	Prince of Wales-Hyder
Zip	Qualitative	Zip code of customer's residence per the billing statement	99927
Lat	Qualitative	Latitude portion of GPS coordinate of customer's residence per the billing statement	56.251
Lng	Qualitative	Longitude portion of GPS coordinate of customer's residence per the billing statement	-133.37571

Column Header	Data Type	Description	Example Data
Population	Quantitative	Population within 1-mile radius of customer, per census data	38
Area	Qualitative / Nominal	Area type of customer’s residence, per census data	Urban
Timezone	Qualitative / Nominal	Timezone of customer’s residence, per customer’s input at signup	America/Sitka
Job	Qualitative	Job of the customer, per customer’s input at signup	Environmental health practitioner
Children	Quantitative / Discrete	Number of children in customer’s household, per customer’s input at signup	1
Age	Quantitative / Discrete	Age of customer, per customer’s input at signup	68
Education	Qualitative / Nominal	Highest degree earned, per customer’s input at signup	Master's Degree
Employment	Qualitative / Nominal	Employment status, per customer’s input at signup	Part Time
Income	Quantitative / Continuous	Annual income of customer, per customer’s input at signup	28561.99
Marital	Qualitative / Nominal	Marital status of customer, per customer’s input at signup	Widowed
Gender	Qualitative / Nominal	Customer’s gender identity, per customer’s input at signup	Male
Churn	Qualitative / Ordinal	Yes or No, Customer has churned in the last month	No
Outage_sec_perweek	Quantitative / Continuous	Average (summary statistic) of outages in the customer’s neighborhood, in seconds	6.972566093
Email	Quantitative / Discrete	Number of company emails sent to customer in the last year for marketing or correspondence	10
Contacts	Quantitative / Discrete	Number of times customer contacted technical support	0
Yearly_equip_failure	Quantitative / Discrete	Number of times a customer’s equipment failed, resulting in reset/replacement	1
Techie	Qualitative / Ordinal	Yes or No, survey question on customer’s self-identity as technically inclined	No
Contract	Qualitative / Ordinal	Length of time for customer’s contract term by category	One year
Port_modem	Qualitative / Ordinal	Yes or No, Customer has a portable modem	Yes

Column Header	Data Type	Description	Example Data
Tablet	Qualitative / Ordinal	Yes or No, Customer owns a tablet device	Yes
InternetService	Qualitative / Categorical	Type of internet service	Fiber Optic
Phone	Qualitative / Ordinal	Yes or No, Customer has phone service	Yes
Multiple	Qualitative / Ordinal	Yes or No, Customer has multiple lines of service	No
OnlineSecurity	Qualitative / Ordinal	Yes or No, Customer has an online security add-on	Yes
OnlineBackup	Qualitative / Ordinal	Yes or No, Customer has an online backup add-on	Yes
DeviceProtection	Qualitative / Ordinal	Yes or No, Customer has a device protection add-on	No
TechSupport	Qualitative / Ordinal	Yes or No, Customer has a technical support add-on	No
StreamingTV	Qualitative / Ordinal	Yes or No, Customer has streaming TV	No
StreamingMovies	Qualitative / Ordinal	Yes or No, Customer has streaming movies	Yes
PaperlessBilling	Qualitative / Ordinal	Yes or No, Customer has paperless billing	Yes
PaymentMethod	Qualitative / Nominal	Type of payment method for service	Credit Card (automatic)
Tenure	Quantitative / Continuous	Number of months a customer has stayed with the provider	6.795512947
MonthlyCharge	Quantitative / Continuous	Average monthly charge customer has received	171.4497621
Bandwidth_GB_Year	Quantitative / Continuous	Average amount of data used in GigaBytes (GB)	904.5361102
item1	Qualitative / Ordinal	Survey question re: timely responses by provider	5
item2	Qualitative / Ordinal	Survey question re: timely fixes by provider	5
item3	Qualitative / Ordinal	Survey question re: timely replacements by provider	5
item4	Qualitative / Ordinal	Survey question re: reliability of provider	3

Column Header	Data Type	Description	Example Data
item5	Qualitative / Ordinal	Survey question re: options available by provider	4
item6	Qualitative / Ordinal	Survey question re: respectful response by provider	4
item7	Qualitative / Ordinal	Survey question re: courteous exchange by provider	3
item8	Qualitative / Ordinal	Survey question re: evidence of active listening by provider	4

C1. Plan to Assess Quality of Data

To assess the quality of the data, I import the churn_raw_data.csv file into Jupyter Notebooks and use the pandas package to convert it into a DataFrame for easy tubular inspection.

After using the `pd.read_csv()` method to create the DataFrame, I use the `.describe()`, `.shape`, and `.info()` methods and attributes to start exploring the data and return some initial summary statistics.

From there, I use the following plan to detect issues with the data: detect duplicates, detect missing values, detect outliers, and detect variables that need to be re-expressed as a different variable type. I also check the titles of each column header for Python formatting in snake case, with lowercase letters and words separated with underscores.

To detect duplicates, I use `df.duplicated()` to return a Boolean Series where `True` values indicate the duplication of a complete row previously found. Chaining the `.value_counts()` method to `.duplicated()` will return a count of how many times each row is found duplicated in the DataFrame.

To detect missing values, I use the `df.isnull().sum()` method to check for any missing values and how many there are. I pair this further with `df['col'].nunique()` on each column to return a list of all the unique responses for each column, which can assist with the next part of the process, identifying outliers.

To detect outliers, I use a combination of histograms and box plots to look for data distributions and outliers. I also check how far outside the Interquartile Range (IQR) the outliers are and create upper and lower limit thresholds.

I check which columns need their types re-expressed with a more appropriate data type and if the values need to be remapped.

Finally, I inspect the column headers to check which ones need to be renamed to follow PEP 8 naming conventions.

C2. Justification of Approach

The churn dataset contains both quantitative data and qualitative data. The findings from the detection stage of the analysis determine how to move forward with the treatment phase of analysis. Using the Data Dictionary and browsing the data via code assists in understanding the context of the data and its intended use.

Duplicate records can distort analysis results, leading to incorrect conclusions about the data. By treating these, we ensure data integrity. Eliminating duplicates improves resource efficiency as duplicates take up additional storage and computational resources. Lastly, duplicates can indicate issues with consistent reporting. For analysis to be accurate, it depends on unique and non-redundant data. There are many instances where having a duplicate value in a column is legitimate, appropriate, and expected, but to have a row that duplicates every value across 51 variables is redundant data.

Missing data has its own set of issues: it lacks completeness that makes a dataset fully representative of what it's measuring, it allows for introducing (or further implementing) bias and inaccuracy, and it reduces the performance of machine learning models. Still, it’s often better to impute data for missing values than to drop any rows or columns with missing data unless the amount of missing data is substantial. More data is typically considered safer for analysis than less data.

When treating missing values, checking for distribution and outliers will help determine if imputation is the best option and, if so, what value should be imputed (e.g., mean, median, mode).

The findings from this step will provide insight on how to treat the missing data as distributions should be maintained. Three primary methods to treat outliers are to leave them alone, delete them, or replace the values. Replacing the values may replace extremes with a measure of center (e.g., mean or median) or detecting a floor or ceiling value based on a percentile.

From a cursory glance, several qualitative variables likely need to be re-expressed as categorical variables. If the variables are “yes” or “no” responses, it will likely be best to modify them to be ordinal categories where “no” is equivalent to 0 and “yes” is equivalent to 1.

The change in data type is essential as some of the information is categorical, such as identifying the state of residence. Some values indicate whether or not a customer has utilized a specific service. Others are ordinal, where the records indicate a scale at which a customer responded to a survey question.

Updating the column names to reflect Python best practices creates a more consistent reading experience. While not always favored by an analyst looking through spreadsheets, the Pythonic naming convention makes using column names in DataFrame analysis via pandas easier.

C3. Justification of Tools

I am choosing to use Python over R as it is the more commonly used language in the industries where I’m interested in working. It’s used internally within my work (even in academia) and is helpful for me to learn when working with our data engineers. I find it easier working in Anaconda/Jupyter Notebooks and/or Google Colab than I do in R Studio. While R is a valuable language, having industry-level experience and proficiency in Python will open more opportunities for data science roles. At present, more machine-learning tools are built with Python in mind.

I use the following packages: Pandas, NumPy, Matplotlib, Seaborn, Scikit-Learn, and the SciPy stats package.

Pandas is beneficial because it structures the data in tabular form (DataFrames and Series) with many built-in methods for functionality to quickly explore and understand the data (including missing data) via summary statistics and data type identification. It also allows easy importing of the raw data and exporting the cleaned data back to CSV format, which is very handy for analysts working with the data in no-code environments.

NumPy offers additional mathematical computation functionality, including detecting and imputing median and mode. It is helpful regarding measures of center and spread and performing statistical testing on the data.

Matplotlib and **Seaborn** are data visualization packages. Matplotlib offers some simplistic yet powerful capabilities, like plotting histograms to understand data distributions. Seaborn offers added functionality by including boxplots, which help identify quartiles, IQR, and outliers.

The **stats** package from **SciPy** builds on the ability to detect medians and modes and adds helpful methods for data imputation. This practice is useful for treating missing data and detecting and treating outliers where values may need to be replaced.

Scikit-Learn provides the **PCA** module, which allows the performance of principal component analysis. This dimensionality reduction technique helps pick out the most influential and impactful numeric variables across a dataset with many features/columns.

C4. Data Cleaning Code (Detection Only)

See attached file (Murdock_Shanay D206 PA Churn.ipynb) for code.

Note: While there are more code-efficient ways of performing this analysis, I chose to take a more granular approach with each variable and look at each in more detail than I could by pairing functions or loops to prevent so much repeated code.

D1. Cleaning Findings

General Issues

- 52 `State` values were found. This dataset includes Washington D.C. (DC) and Puerto Rico (PR).
- `Zip` has 773 values with zip codes featuring only 3 or 4 digits. This is possible in East Code zip codes, but the preceding 0s must be retained as significant values.
- Column names do not match a consistent naming convention.
- There are too many time zones to represent the United States appropriately.

Duplicates

- No duplicate rows were found. All 10,000 rows are unique and should be retained.

Missing Values

- `Children`, `Age`, `Income`, `Techie`, `InternetService`, `Phone`, `TechSupport`, `Tenure`, and `Bandwidth_GB_Year` all have missing values. Treatment of these will depend on their distribution.

Outliers

- It is worth remembering that `Population` is not of the city of residence but within a 1-mile square radius. These outliers feel particularly high for that. However, the city with the highest frequency in the dataset is Houston, which has a population of over 2.3 million. According to a Google search for the highest population densities per square mile in the United States (which is still more land coverage than a 1-mile radius), approximately 63,000 people per square mile is the highest population density in the U.S. I have questions as to whether all the data is reported correctly. Still, perhaps it's worth creating a 90th percentile ceiling and imputing that data for any values currently above the 90th percentile. This will be explored further in D2.
- `Children` treats any number above 7 as an outlier but these are not outside the logical bounds and can remain as is.
- `Income` features many outliers, but these do not seem outside of logical bounds as income distribution in the United States is vast. Plus, the largest number of customers come from major cities where higher incomes are typically found. However, 7 very high outliers change the distribution drastically.
- `Outage_sec_perweek` count any data point greater than 19 as outliers. There are 514 values above 19, so extreme outages may have occurred. These are not illogical, though, so the outliers can remain untreated. Eleven values are recorded as less than 0; this is not possible as this data represents the absence of service. These values will need to be treated with the `abs()` function to remove their sign and make all the values positive.
- `Email`, `Contacts`, `Yearly_equip_failure`, `MonthlyCharge`and each of the survey questions feature outliers. All of the outliers are within logical or expected bounds.

Change Data Type

- `Zip`, `Lat`, and `Lng` should be re-cast as string/objects to retain precision as geographical identifiers.
- The following variables should be recast as nominal categories: `Area`, `Timezone`, `Job`, `Employment`, `Marital`, `Gender`, `Contract`, `PaymentMethod`, and `Education`.
- The following variables should be recast as ordinal categories: `item1`, `item2`, `item3`, `item4`, `item5`, `item6`, `item7`, `item8`.
- The following variables should be recast as Booleans: `Churn`, `Techie`, `Port_modem`, `Tablet`, `InternetService`, `Phone`, `Multiple`, `OnlineSecurity`, `OnlineBackup`, `DeviceProtection`, `TechSupport`, `StreamingTV`, `StreamingMovies`, `PaperlessBilling`.

D2. Justification of Mitigation Methods

General Issues

- The names of the column headers should follow a consistent format (and some could have their meaning clarified). I replace them with a snake_case format.
- The columns associated with the survey questions will be renamed to clarify what the question is measuring.
- I use a Pandas Series method called `zfill()` to add preceding `0`s to the ZIP codes that have less than 5 digits. It is common on the East Coast for ZIP codes to start with 0 and there are 773 instances in the dataset. Due to the nature of CSVs, any data type cast as an integer will drop leading 0s, not recognizing them as statistically significant.
- The `Timezone`s overlap by having too much detail of subcategories. I simplify this structure by merging the subcategories so that only the high-level categories persist. For instance, "America/Toronot" and "America/New York" represent the same time zone. I will change that to "US/Eastern".
- The `Outage_sec_perweek` column contains erroneous negative values that must be corrected. I use the `abs()` function to flip the numbers of the values below 0 from negative to positive. This does not change the distribution of data.

Duplicates

- There are no duplicate records in this dataset. No treatment is needed as all 10,000 rows are unique.

Missing Values

- `Children`, `Age`, `Income`, `Techie`, `InternetService`, `Phone`, `TechSupport`, `Tenure`, and `Bandwidth_GB_Year` all have missing values.
  - `Children` is missing 2495 values, approximately 25% of the data. The column will be imputed with `0`. This is a self-reported value by customers, and if they choose not to report it, the safest answer is `0`. This is also the statistically relevant solution because `0` represents the median and mode of the existing data.
  - `Age` is missing 2475 values, approximately 25% of the data. This value is self-reported by the customer. If this is necessary information, it should become required information in any sign-up forms. The column has a fairly uniform distribution, so a rounded mean (integer) would be best to impute for the missing values. It will cause a spike in the center due to the number of missing values but will not change the overall distribution.
  - `Income` is missing 2490 values, approximately 25% of the data. This value is self-reported by the customer. If this is necessary information, it should become required information in any sign-up forms. Based on the right skew of the data, removing the highest seven outliers (everything above $176,000), rechecking the median, and imputing the new median would likely create the most representative data. With so many data points missing, it will cause a spike, but the distribution will remain steady.

        - `Techie` is missing 2477 values, approximately 25% of the data. When dealing with Boolean values, it's safer to impute `"No"` or `0` than to assume the affirmative.
        - `InternetService` is missing 2129 values, approximately 22% of the data. I fill this in with `"None"` as it's possible that a customer might not have internet service but may have other services with the company."
        - `Phone` is missing 1026 values, approximately 10% of the data. When dealing with Boolean values, it's safer to impute `"No"` or `0` than to assume the affirmative.
        - `TechSupport` is missing 991 values, approximately 10% of the data. When dealing with Boolean values, it's safer to impute `"No"` or `0` than to assume the affirmative.
        - `Tenure` is missing 931 values, approximately 9% of the data. As this is an internal calculation, the process for creating/inputting this data should be reviewed. The distribution is bimodal, so I fill the missing values with the mode.
        - `Bandwidth_GB_Year` is missing 1021 values, approximately 10% of the data. As this is an internal calculation, the process for creating/inputting this data should be reviewed. The distribution is bimodal, so I fill the missing values with the mode.

Outliers

- It is worth remembering that this `Population` is not of the city of residence but within a 1-mile square radius. These outliers feel particularly high for that. However, the city with the highest frequency in the dataset is Houston, which has a population of over 2.3 million. According to a Google search for the highest population densities per square mile in the United States (which is still more land coverage than a 1-mile radius), approximately 63,000 people per square mile is the highest population density in the U.S. I have questions as to whether all the data is reported correctly. Still, perhaps it's worth creating a 90th percentile ceiling and imputing that data for any values currently above the 90th percentile.
- While high, the outliers in the `Children` column are not outside of logical bounds and can remain untreated.
- `Income` has many outliers, but these do not seem outside of logical bounds as income distribution in the United States is vast. Plus, the largest number of customers come from major cities where higher incomes are typically found. However, 7 very high outliers change the distribution drastically.
- `Outage_sec_perweek` count any data point greater than 19 as outliers. There are 514 values above 19, so extreme outages may have occurred. These are not illogical, though, so the outliers can remain untreated. Eleven values are recorded as less than 0; this is not possible as this data represents the absence of service. These values must be treated with the `abs()` function to remove their sign and make all the values positive.
- `Email`, `Contacts`, `Yearly_equip_failure`, `MonthlyCharge`, and each of the survey questions feature outliers. All of the outliers are within logical or expected bounds. For that reason, these outliers will not be treated and are left as is.

Change Data Types

- I change the data types of the `Zip`, `Lat`, and `Lng` columns from integer/float to string to ensure precision and data integrity as these are qualitative pieces of information that happen to take numeric values. We are not interested in finding summary statistics on these values and they express geographic locations.

- I change the data type of `Children` from `float64` to `int64` as all the values should be integers. The existing values all have a `.0` value so there is no distribution change in the data.
- I change `Area`, `Timezone`, `Job`, `Education`, `Employment`, `Marital`, `Gender`, `InternetService`, `Contract`, and `PaymentMethod` to nominal variables. These are categories that feature a low number of unique values and lack scale/order/rank in the values.
- I change `Churn`, `Techie`, `Port_modem`, `Tablet`, `Phone`, `Multiple`, `OnlineSecurity`, `OnlineBackup`, `DeviceProtection`, `TechSupport`, `StreamingTV`, `StreamingMovies`, and `PaperlessBilling` to the Boolean type based on the presence of only two variables, representing "Yes" and "No". Zero (`0`) will be used for "No," and one (`1`) will be used for "Yes".
- I change `item1`, `item2`, `item3`, `item4`, `item5`, `item6`, `item7`, and `item8` to ordinal categorical variables based on the low number of unique values and the necessity of scale/order/rank in the values.

D3. Summary of the Outcomes

The first thing I did was change the names of the column headers to follow Pythonic conventions. This makes it easier to read and understand what the columns represent.

Next, I collapsed the time zone categories into the standard time zones across the United States. This removed an unnecessary level of precision and duplicate values. While these are not “duplicates” in the sense we’re talking about, “America/Indiana/Knox” and “America/Chicago” represent the same time zone.

I treated the negative values found in the `outage_sec_per_week` column as only positive values should be found to record the absence of service. This changed the minimum value without changing any of the measures of center.

`children`, `age`, `income`, `internet_service`, `techie`, `phone_service` , `tech_support`, `tenure`, and `bandwidth_gb_year` all had values imputed to fill the missing values. This brought their count of records up to 10,000, along with the rest of the columns. For some, a mean, median, or mode was imputed. For others, a “No” or “None” value was added depending on the context. This ensured there was no missing data and that the data added supported the existing data distributions.

The only column with extreme outliers that didn’t make logical sense was the `population` column, as there was such a large jump from the 75th percentile (13,168) to the max value (111,850). I capped the maximum off at the 90th percentile and imputed that value for any outlier above that. That ensured there remained values for those points.

Lastly, I changed several columns' data types to reflect their purpose better. This helps ensure consistency in adding new data and analyzing existing data. It helped to retain the precision of data being lost to generic data types in CSV format and should retain the necessary level of precision going forward.

D4. Data Cleaning Code (Treatment)

See attached file (Murdock_Shanay D206 PA Churn.ipynb) for code.

D5. Copy of Cleaned Code

See attached CSV file ('churn_data_cleaned_Murdock_Shanay.csv') for cleaned data.

D6. Summary of Limitations

A limitation is that the collapse of time zones is being done in July in Eastern Daylight Time and does not account for changes in the time zones due to Daylight Savings Time depending on the time of year. This would typically be automated using the datetime data type but would also feature more corresponding date data.

The dataset was missing values from 9 columns, ranging between about 900 to 2,500 missing values for each of those variables. At a dataset of 10,000 rows, this isn't unacceptable. Still, it would be helpful to know if those values are missing completely at random (MCAR), missing at random (MAR), or missing not at random (MNAR), as that might indicate why the data is missing. Some of the values are self-reported values through customer sign-up information, and depending on the input (paper versus web form), these could be made as required inputs in the future.

Imputation was decided based on a column's distribution. Still, a limitation is that as this is a solo project using fictionalized data, there isn't someone with institutional knowledge to try to glean further insights.

Only one outlier was seen as extreme enough to change. The population data seemed very high for anything to do with a 1-mile radius. The challenge was deciding where to place an arbitrary cap at a percentile and whether to impute the cap or median data. I chose the cap to keep the data points in line with their originals, even though it created a large spike at the end of the distribution when seen in a histogram.

I chose not to drop any outliers to preserve as much data as possible for further analysis.

I also made judgment calls on re-expressing some of the columns as categories. For instance, education and marital data could be changed to nominal or ordinal data, depending on how the data is used. At this time, I chose to leave them as nominal categories.

D7. Impact of Limitations

While the dataset is now fully complete (there are no missing values), the imputations of data were best guesses. This will introduce some bias into the dataset as we cannot turn to the data owner and ask why the data is missing, if it can be retrieved, or the implications of not having it.

There are new spikes in the distribution, which another analyst would see if EDA is performed. While we can hope that they'd understand imputation happened in certain distributions (e.g., age, population), that's not a given, and documentation would need to be added to ensure understanding for proper collaboration.

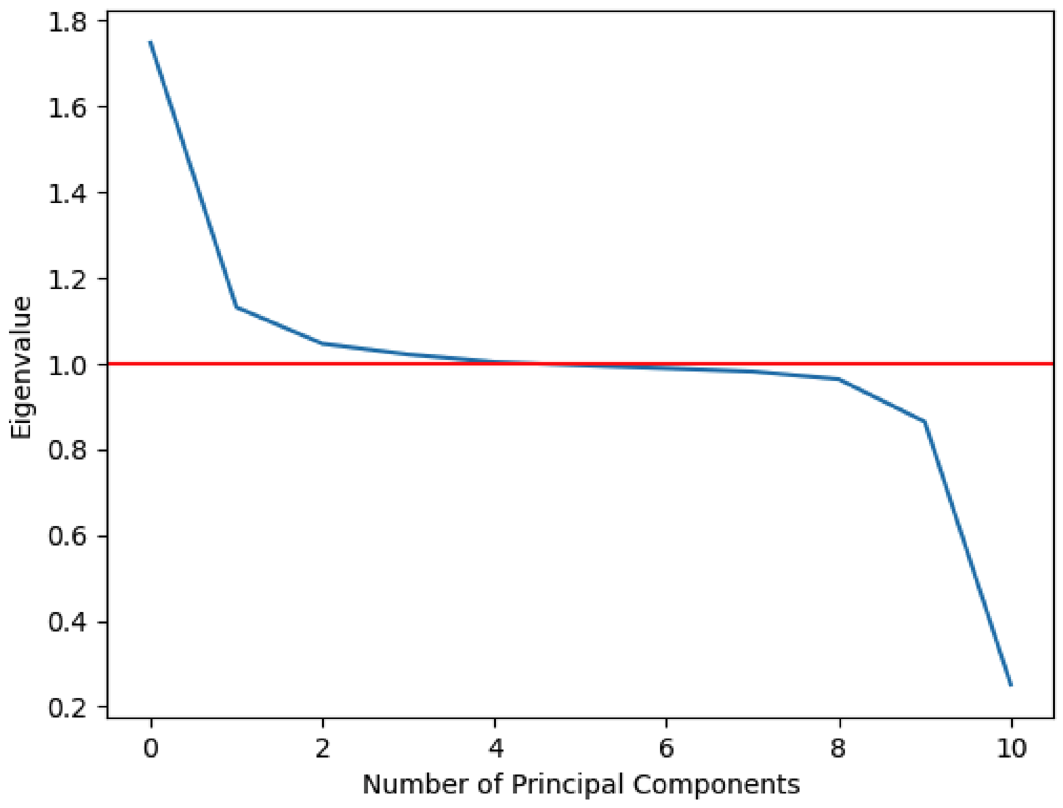
The imputations also have downstream consequences, especially in the demographic data. Nearly 25% of the customers now appear to be age 53, and rather than a tail of high-population cities, there is a giant spike at the very end, which can be misleading.

E1. Principal Components

(DataCamp, 2024, Dimensionality Reduction in Python)

The following variables were used for Principal Component Analysis as they are continuous variables: `population`, `children`, `age`, `income`, `outage_sec_per_week`, `email`, `contacts`, `yearly_equipment_failure`, `tenure`, `monthly_charge`, `bandwidth_gb_year`.

Below is a screenshot of the scree plot generated in Jupyter Notebook.



`PC1`, `PC2`, `PC3`, and `PC4` meet the Kaiser Criteria.

E2. Criteria Used in PCA

`PC1`, `PC2`, `PC3`, and `PC4` meet the Kaiser Criteria.

The first four Principal Components meet or exceed the criteria of having an eigenvalue of 1.0 or more. The elbow of the scree plot also levels off after the 4th component. Realistically, based on how close `PC4` is to 1.0, it only just meets the minimum criteria so in a professional setting, I would confer with a team member or leader to determine if all 4 should be kept or if only the first 3 should be kept until having a deeper understanding of the data.

E3. Benefits of Using PCA

While it is generally agreed that having too much data is better than the alternative of having too little data, the amount of data can become cumbersome, and it can be harder to derive insight from the raw data or know what variables to compare for trends and insights.

With large datasets comes enormous complexity. Principal Component Analysis is one unsupervised learning technique for dimensionality reduction. The process allows us to break the dataset down into smaller groupings of related continuous variables (and reducing the amount of redundant data) while dismissing the groupings that have less impact or insight on any patterns that can be found (Dimensionality Reduction in Python, DataCamp, 2024).

Principal Component Analysis also provides thresholds for understanding how well each principal component represents the data through reduction. This threshold allows the user to understand what principal components are worth keeping for insight.

This process is generally beneficial for datasets with many variables (dimensions). Still, as the Churn dataset is mainly made up of qualitative or categorical variables, it is worth noting that only about 20% of the dataset's dimensions can be used in the PCA model, with only 11 dimensions being continuous data types.

F. Panopto Video Walkthrough

https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=958f343e-00b2-4704-b654-b1ae00cbc983

G. Code Resources

Boeye, J. (2024). *Dimensionality Reduction in Python*. https://app.datacamp.com/learn/courses/dimensionality-reduction-in-python

McCoy, S. (2024). *Murach's Python for Data Science* (2nd ed.). Murach.

McKinney, W. (2022). *Python for Data Analysis* (3rd ed.). O'Reilly Media, Inc.

pandas (2024). *Categorical data*. Pandas Documentation. https://pandas.pydata.org/pandas-docs/stable/user_guide/categorical.html

pandas (2024). *Pandas.Series.Str.Zfill*. Pandas Documentation. https://pandas.pydata.org/docs/reference/api/pandas.Series.str.zfill.html

Walker, M. (2024). *Python Data Cleaning Cookbook* (2nd ed.). Packt Publishing.

WGU Course Ware videos (2024).

H. Other Resources

Grus, J. (2019). *Data Science from Scratch: First Principles with Python* (2nd ed.). O'Reilly Media, Inc.

Raschka, S., Liu, Y., & Mirjalili, V. (2022). *Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python*. Packt Publishing.

timeanddate (2024). *Time Zones in the United States*. Retrieved June 5, 2024, from https://www.timeanddate.com/time/zone/usa