NUM3 Task 1: Data Cleaning

Steven Oldenburg

Western Governors University

D206 Data Cleaning

01/31/2024

NUM3 Task 1: Data Cleaning

A: Question or Decision

What variables determine a customer's churn and what variables can identify customers who are more likely to continue service with our company?

B: Identifying Data

The raw data contains 10,000 rows and 52 columns. I used Python to export the column names and data types then filled in information using the .pdf file from the scenario and the raw data.

Column Name	Data Type	Description	Data Example
Unnamed: 0	int64	Unused column	The data matches the Case Order column
CaseOrder	int64	Placeholder for the original index	Each row is numbered in this fashion to help organize the data
Customer_id	object	Unique customer-id	7 digits starting with a letter unique to each customer
Interaction	object	A type of transaction code	Each transaction is given an interaction code which appears to be a randomly generated number
City	object	The city the customer lives in	The data lists cities from all over the country such as "Del Mar"
State	object	The State the customer lives in	States are listed in the data abbreviated to two characters such as "AK" for Alaska
County	object	The county the customer lives in	The data lists counties from all over, such as "Scott" county
Zip	int64	An integer which is stored to identify the zip code	The data contains 5 numbers such as 99927
Lat	float64	Latitude coordinates	Latitude coordinates such as "56.251"
Lng	float64	Longitude coordinates	Longitude coordinates such as "-133.376"

Population	int64	Population within a mile of the customer	Population amounts data such as 50079
Area	object	Categorized density descriptor	This describes "Rural", "Urban" or "Suburban" density
Timezone	object	Customer time zone categorized	Categorized time zones such as "America/Sitka"
Job	object	Categorized information about the customer's job	Jobs are listed in the data such as "Solicitor"
Children	float64	Number of how many children the customer has	Children's information ranges between 1-10
Age	float64	Customers age	A number representing the customer's age such as "23"
Education	object	Customers education attainment	The columns list categories of educational attainment such as a "master’s degree"
Employment	object	Categorized information about how much a customer works	Categorized information such as "unemployed" or "part-time"
Income	float64	Customer yearly income information	A dollar amount of yearly income such as 28561.99
Marital	object	Customer marital status	Categorized status such as "widowed" or "single"
Gender	object	Customer gender	Categorized gender as "male" or "female"
Churn	object	Customer churn data	Categorized as "Yes" or "No"
Outage_sec_perweek	float64	Customer Outages in seconds per week in the neighborhood	This is given as a number averaged by seconds per week
Email	int64	This is a number of emails sent to the customer	Numbers appear to range from 1-23
Contacts	int64	How many contacts the company has had with the customer	Contacts range from 0-7, so it cannot be how many people are listed on the service
Yearly_equip_failure	int64	How many pieces of hardware have failed per year	An integer that ranges from 0 - 6
Techie	object	Self-reported customer tech savviness	Categorized as "Yes" or "No", doesn't explain exactly what it is
Contract	object	If the customer has a contract with the company	Contracts are categorized into three categories

Port_modem	object	If the customer has a portable modem	Categorized by "yes" or "no" if the customer has one
Tablet	object	If the customer has a tablet	Categorized by "yes" or "no" if the customer has one
InternetService	object	Describes the internet service a customer has	Categorized by service, there are three types such as DSL
Phone	object	If the customer has phone service with the company	Categorized as "yes" or "no" in the data
Multiple	object	Most likely if the customer has multiple phone lines	Categorized as "yes" or "no" in the data
OnlineSecurity	object	If the customer has security services such as anti-malware with the company	Categorized as "yes" or "no" in the data
OnlineBackup	object	If the customer has an online backup service	Categorized as "yes" or "no" in the data
DeviceProtection	object	If the customer has a device protection service	Categorized as "yes" or "no" in the data
TechSupport	object	If the customer has tech support services	Categorized as "yes" or "no" in the data
StreamingTV	object	If the customer has a steaming TV service	Categorized as "yes" or "no" in the data
StreamingMovies	object	If the customer has a steaming movie service	Categorized as "yes" or "no" in the data
PaperlessBilling	object	If the customer has paperless billing	Categorized as "yes" or "no" in the data
PaymentMethod	object	Categorizes the customer's payment method	Categorized by type such as "credit card"
Tenure	float64	Months as a customer	A number ranging from 1 to 71
MonthlyCharge	float64	The monthly charge for services	A number ranging from 77 to 315
Bandwidth_GB_Year	float64	Yearly bandwidth in GB	A number from 155 to 7158
item1	int64	Timely response	Survey data ranging 1 - 8
item2	int64	Timely fixes	Survey data ranging 1 - 8
item3	int64	Timely replacements	Survey data ranging 1 - 8
item4	int64	Reliability	Survey data ranging 1 - 8
item5	int64	Options	Survey data ranging 1 - 8
item6	int64	Respectful response	Survey data ranging 1 - 8
item7	int64	Courteous exchange	Survey data ranging 1 - 8

item8	int64	Evidence of active listening	Survey data ranging 1 - 8

C1: Plan to Clean Data

First, I will keep a data backup. Second, load the data into Python using the read_csv() function in Pandas. Third, I will evaluate the data structure using functions or attributes such as .shape or .dtypes. These two attributes explain the data frame, .shape counts the rows and columns while .dtypes returns column names with data types such as integers or objects. I also used the .unique() attribute for each column which returns all the different possible values in that column. Fourth, I will rename data for different spellings and rename the survey data columns, so they are much more understandable. Fifth, I will look for missing data such as "NA" or "null" values, I will pay particular attention to see if the data is missing at random or not. Sixth I will look for outliers that may skew the data. Finally, I will input data values using the mean, median, or mode depending on what data needs to be imputed.

C2: Justification of Approach

This approach gives me a methodical way to make sure I cover everything that is needed to truly clean the data. As of right now, I have only mildly reviewed that data, I plan to let Python do the heavy work. This approach can be summed up as this, learn the data structure, rename the data to standardize it, find what is missing, how it's missing, look for skewed data, and input data to standardize null values so that Python can do the heavy math. Characteristics of the data include that it is very organized out of the box, some data types may not be the best data type for what we are trying to accomplish, and some data is labeled poorly. There is quite a lot of data missing in a subset of the data. About 2500 records are missing some information in their rows and much of that is overlapping. This data looks to be missing not quite at random because the entries with missing data overlap, meaning the source itself is most likely missing the data however it was acquired. On the positive side, I observed no particular data gaps, meaning it was at random within its subset and didn’t seem to be selective as a group that would skew the data, for instance, if only the data was missing from the older population. I used the following functions/attributes:

read_csv(): to load the data

.shape: to give me a count of rows and columns

.dtypes: to see if numbers are integers and should they be or if they are strings instead.

.unique(): to see all the values that can be returned for each column

.head()): to see how the data looks

duplicated().any(): to check for duplicates

.isna().sum()): to count Na, NaN, or Null values to see how much is missing and where

.mean(): this finds the average

.median(): this finds the middle number of the data

find_IQR(): is a function that finds the interquartile range

count_df(): is a function that counts yes as 1 and no as 0

.plt.hist(): plots histograms

.boxplot(): plots boxplots

C3: Justification of Tools

My main tool will be Python, I'm going to be using this in the Jupyter Notebook environment that was installed from the program Anaconda. I'm using this because it's a great environment that is easy to use. Anaconda functions as a type of data tool launcher that helps organize packages. Jupyter Notebook functions as a very user-friendly Python environment. It allows me to write code in an environment like Notepad and provides details or hints with color codes. Another reason I'm using this is because it's something I'm already familiar with and data cleaning is something new. I also plan to use packages such as Numpy and Pandas. Numpy is great for working with arrays and Pandas is good for loading data sets. I will also use Mathplotlib to plot charts. I created a function find_IQR() to find the interquartile range of data that doesn't have a normal distribution to find outliers. I also created a function count_df() which just counts "yes" or 'no" as 1 or 0 so I can better assess some categorial information when needed. I'm using Timezonefinder to sort timezones in conjunction with the Datetime package. I also used Fuzzywuzzy to assess the quality of the city names.

C4: Provide the Code

See code attached as "Describe.ipynb"

D1: Cleaning Findings

I found no duplicate data as the duplicated().any function/attribute returned false. I did find quite a lot of missing data. The missing data is as follows:

Column	Missing records
Children:	2495
Age:	2475
Income:	2490

Techie:	2477
Phone:	1026
tech_support:	991
tenure:	931
bandwidth_gb_year:	1021

Zip codes were integers instead of strings. Columns and row values did not follow any standardized naming or capitalization conventions. Time zones were difficult to read, for instance, the first time zone is “America/Sitka” which unless someone knows Sitka is in Alaska will not help them assess the time zone. The job list is massive and follows no standard conventions. Education is mixed with lowercase and upper also some categories overlap. Multiple married values overlap as well as some capitalization issues. City names were also misspelled, but it’s a massive list, and would someone manually assess each city to make sure which spelling is correct or a reference city name list using the zip code? Doing this is outside the scope of the research question and exercise. Assessing the missing data using graphs showed most distributions are not normal distributions. I chose to calculate the outliers using the interquartile range, however, a small assessment made it apparent should not exclude the values because the research question relates to the churn rate. The outliers in the case of children for very large households showed a tendency to churn less which is part of the research question. The outliers for income held a slightly higher churn rate. Monthly payments had 5 outliers on the extreme payment side, I chose still to include these because they related directly to the research question, and all 5 of them churned. It only made sense to keep information that payments that are too high may cause customers to churn given the research question.

D2: Justify Methods

The first thing I did was change the entire dataset into lowercase and set up standardized column names. I chose to use the snake case because it would be easier to read. Setting the whole dataset as lowercase also eliminates any extra mitigation, I would need to do about capitalization issues. For instance, “Married” and “married” would be instantly merged. I also dropped the “unnamed: 0” column as it was no longer being used or provided any value to the dataset. I changed the zip code to a string datatype just in case it’s needed in the future to be used categorically to possibly identify city names. I rounded the income column to 2 decimal places just in case there was an issue in the dataset. I created a new data frame containing missing values just in case it's needed to compare with the original or to help identify missing values that need to be changed. I ran a Fuzzywuzzy similarity search on the city names which identified quite a few misspellings, however, without a type of reference I couldn’t change the city names because I don’t know the correct spelling, and it's outside the scope of the churn rate question. I first tried to change time zones based on a zip code index, but this did not work very well, and wasted some time. However, I was able to identify a way and reason the latitude and longitude functions were present as they were used to time the listed time zone. I used the same time zone function to use a standardized version such as PST or EST that simplified the categories in that field which made better business sense, especially if the company plans to call the customer so they can quickly identify their time zone and know when it is illegal to call. I changed the grouping of education to merge some college with high school as it just made sense to simplify the information, so it doesn’t overlap. I also did the same to the martial status and set it to married or unmarried getting rid of special categories such as widowed and divorced. With all the columns missing I assessed them and chose to use the median values in all of them because they are categorial values and would make little sense to place a large number of values in the opposite category of the norm when there are only 2 values to choose from. The only exception to this is the children category, I choose to use the mean rounded down and set it to 2 children. The reason I choose to do this is that it would skew the data even further to one side if I placed 2500 entries on top of less than 2000 entries already. Another reason I chose to set it as 2 is that I found the churn rate for 2 children to be almost the same average as the churn rate of the records with missing children's values, meaning it wouldn't significantly change the outcome of the research question and if I chose to use the median it would. I chose not to drop any outliers because their values would directly impact the research question and it didn't make sense to drop things such as very high month payments leading customers to churn and this information may be valuable to the bottom line. The method I used to identify these outliers in most cases was to calculate the interquartile range, especially since distributions were not normal distributions. All in all, it was very easy to clean up the dataset, although it was time-consuming I couldn't imagine using any other way to clean up 10000 rows of information except for using programming languages such as Python or R.

Summarize the Outcome

Each step of the process I used is summarized below:

1. I never overwrote the original raw data and instead saved a different output copy. This gave me the ability to reset very quickly using the Python language when I made a mistake.
2. In step 2 I loaded the data into Python using the read_csv() function in Pandas successfully.

3. In step 3 I identified the shape of the data including data types, checked for duplicates and unique categories values looking for misspellings or capitalization issues.
4. Step 4 I changed the data into lowercase and put the columns into snake case standardization. I also started to merge different values in the columns, such as married and education. I changed the standard of the values in the time zone field with one that made more sense for anyone reading the data or using it to dial customers.
5. In the 5th step, I took a deep dive into missing data comparing it to the rest of the data to give me an idea of how not to contaminate my own data set.
6. Finally, I identified outliers using the interquartile range, then checked to see if keeping them made sense, dropping them, or changing their values. In the end, I decided to keep them as they made the most business sense and replaced the NA values with the median in every case but one.

D4: Provide the Cleaning Code

Cleaning code attached as “D206Clean.ipynb”

D5: Cleaned Data

Cleaned data is attached as “clean_churn_data.csv”.

D6: Limitations

The biggest limitation of the data cleaning process is we are not able to identify why certain information is missing. For instance, a lot of the rows with missing data had data missing in multiple columns suggesting that it may have been an issue in the intake of the data. Either the information was not gathered or was not transferred to the raw data sheet I received. Another limitation is in the data gathering process, without information as to why information is gathered or presented a certain way, I’m left to guess what would be a better use, for example, the time zones were not in a reader-friendly format. In reality, during my data cleaning process, I only stumbled on the package and function used to generate it because my first try using zip codes did not work. The time zones were based on latitude and longitude which again leaves the question of how that information was gathered. Finally, the last limitation is the lack of business knowledge. If I had a better understanding of what each business unit would use the information for, I could better organize and model the data specifically for those who would use it directly.

D7: Impact of Limitations

The impact of the limitations can be summarized as follows, without the business knowledge of how the information will be used or how it was gathered it makes it difficult to properly organize the data to fit the business needs itself. It's also possible that the missing data does exist somewhere in the company's systems but it was originally gathered it was missing. Another flaw is how it was gathered, latitude and longitude are not something commonly put on telecommunications applications, without knowing how the information is gathered it is difficult to consider what is applicable or what data can be already skewed. For instance, if much of this data is gathered through email or electronic means it's possible the age data is already biased. The age data was a bit strange especially since it seems wherever or however this data was pulled almost each age was evenly distributed.

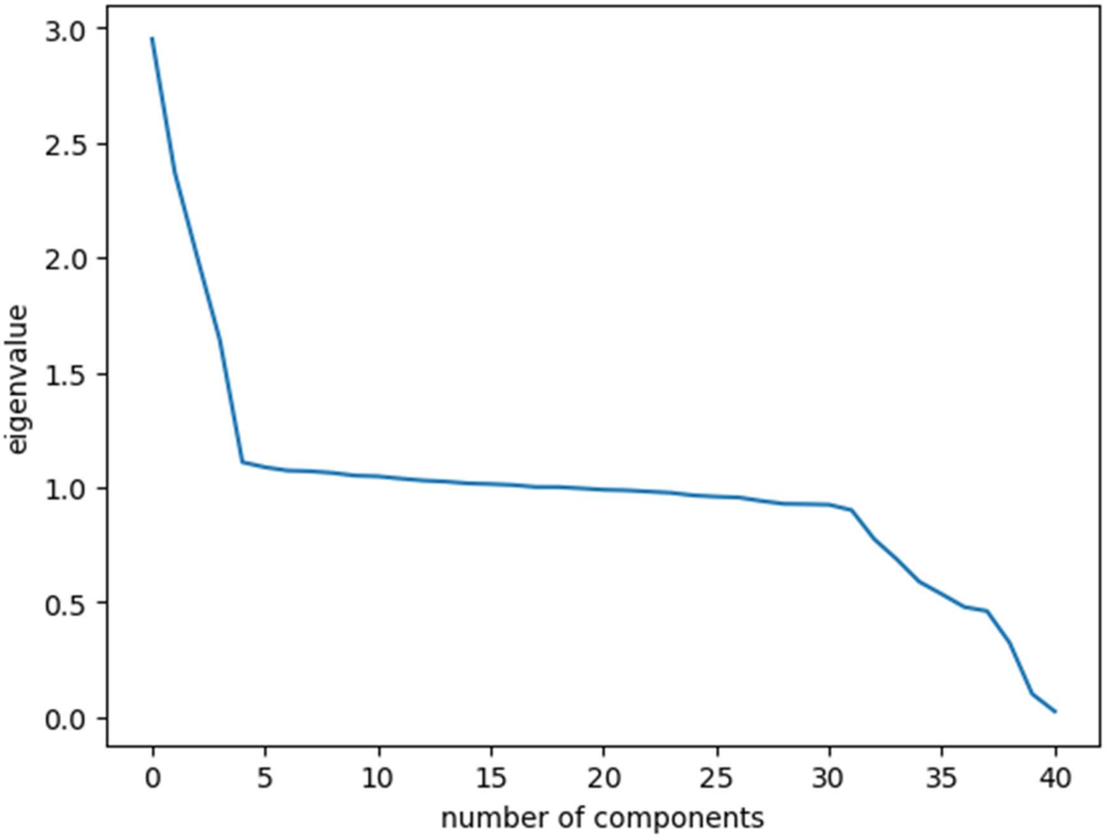
E1: Principal Components

The PCA test I completed used 41 components. Below is the principal component loading matrix:

```
[0.07198052 0.05783948 0.048816   0.04001188 0.0270555  0.02651774
 0.02616467 0.02609362 0.02591445 0.02562672 0.02555049 0.02531604
 0.0251082  0.02499226 0.0248036  0.02473275 0.02462698 0.02441996
 0.02441244 0.02427571 0.02412669 0.0240681  0.02393329 0.02380125
 0.02352274 0.0233947  0.02330456 0.02293875 0.02263189 0.02259836
 0.02253958 0.02196    0.01889188 0.01674991 0.0143598  0.01304002
 0.01167786 0.0112495  0.0078902  0.00245798 0.00060393]
```

E2: Justify Reduced Components

Using the PCA test and scree plot I've determined 24 variables should be used. According to the PCA analysis only using 4 variables account for 22% of the variance in the data frame out of 41 variables. While that sounds great it would lead to too much data loss. After running the scree plot it should somewhere around 20-28 variables are above 1. I was able to calculate a 70% variance ratio using the PCA matrix at 24 variables which is a generally acceptable threshold for PCA tests, and it matched the scree plot for verification. Any further number of components is below the eigenvalue of 1 and should be discarded/reduced. The reason for this is that the variance the subsequent variables are responsible for is so low it would only lead to confusion and would be close to any possible margin of error.



E3: Benefits

The benefit of using a PCA analysis like this is that it helps bring order and structure to the chaos of so many possible variables. That data set had originally 10000 rows and 52 columns which is already difficult to try and visualize what data is the most important. You could pick and choose what you think is the most important, but then it would reflect your bias and I would expect in the business world there are data sets with millions of rows and even more columns. By using this tool, a business can try to find areas or variables to concentrate on, it helps to point to where to look. However, there are some limitations to this test, the dataset provided had a lot of distributions that were not normal. This can lead to attention being given to areas that have little impact. For instance, the largest variance from the data is in yearly equipment failure. Ironically the data shows that customers surprisingly churn less with more equipment failure which makes zero business sense. The test only points to change, things that should be investigated in further detail. Overall using the test was a benefit, it pointed to certain columns I should examine, and it helped me determine that the number one factor I could find in the dataset is its contracts, which just makes sense. Customers in month-to-month contracts are more likely to churn and those in longer contracts significantly churn less. Month-to-month contracts are more likely to churn by %175 over the average of 1- and 2-year contracts.

References

Larose, C. D., & Larose, D. T. (2019). Data science using Python and R. ISBN-13: 978-1-119-52684-1.