Beth Fuller

February 14, 2024

D206: Data Cleaning

Part I: Research Question

A. Research Question

- What factors contribute most to customer churn at the telecommunications company?

B. Variables

Variable Name	Data Type	Description/ Defination	Example (Row 1)
CaseOrder	Qualitative	Sequential placeholder	1
Customer_id	Qualitative	Unique ID for each customer	K409198
Interaction, UID	Qualitative	Unique ID for each interaction (transaction, tech support, sign-up)	aa90260b-4141-4a24-8e36-b04ce1f4f77b
City	Qualitative	City of customer residence	Point Baker
State	Qualitative	State of customer residence	AK
County	Qualitative	County of customer residence	Prince of Wales-Hyder
Zip	Qualitative	Zip code of customer residence	99927
Lat	Qualitative	GPS latitude from customer residence	56.251
Lng	Qualitative	GPS longitude from customer residence	-133.37571
Population	Quantitative	1 mile radius population to customer	38
Area	Qualitative	Rule, Urban, Surburban area based on census data	Urban
Timezone	Qualitative	Time zone based on sign up	America/Sitka
Job	Qualitative	Customer job	Environmental health practitioner

Children	Quantitative	Number of children	NA
Age	Quantitative	Age of customer	68
Education	Qualitative	Highest degree earned	Master's Degree
Employment	Qualitative	Employment status	Part Time
Income	Quantitative	Annual income	28561.99
Marital	Qualitative	Marital status	Widowed
Gender	Qualitative	Male, Female, Nonbinary	Male
Churn	Qualitative	Yes, No - if the customer churned within past month	No
Outage_sec_perweek	Quantitative	Outage seconds (average) per week in customer's neighborhood	6.972566093
Email	Quantitative	Number of marketing or correspondence emails sent to customer in the	10
Contacts	Quantitative	Number of contacts with technical support	0
Yearly_equip_failure	Quantitative	Number of times in past year the customers equipment had to be reset/replaced	1
Techie	Qualitative	Yes, No - if customer views themselves as	No
Contract	Qualitative	Month-to-Month, One Year, Two Year - contract term	One year
Port_modem	Qualitative	Yes, No - portable modem	Yes
Tablet	Qualitative	Yes, No - does customer own	Yes
InternetService	Qualitative	DSL, Fiber Optic, None - internet serivce of customer	Fiber Optic

Phone	Qualitative	Yes, No - phone service	Yes
Multiple	Qualitative	Yes, No - multiple lines	No
OnlineSecurity	Qualitative	Yes, No - online security add-on	Yes
OnlineBackup	Qualitative	Yes, No - online backup add-on	Yes
DeviceProtection	Qualitative	Yes, No - device protection add-on	No
TechSupport	Qualitative	Yes, No - technical support add-on	No
StreamingTV	Qualitative	Yes, No - streaming TV	No
StreamingMovies	Qualitative	Yes, No - streaming movies	Yes
PaperlessBilling	Qualitative	Yes, No - paperless billing	Yes
PaymentMethod	Qualitative	Electronic Check, Mailed Check, Bank (Automatic Bank Transfer), Credit Card	Credit Card (automatic)
Tenure	Quantitative	Number of months customer has been with provider	6.795512947
MonthlyCharge	Quantitative	Amount chared monthly to customer (as	171.4497621
Bandwidth_GB_Year	Quantitative	Amount of data used in GB by year (as average)	904.5361102
item1	Qualitative	Scale of 1-8 importance to customer - Timely response	5
item2	Qualitative	Scale of 1-8 importance to customer - Timely	5
item3	Qualitative	Scale of 1-8 importance to customer - Timely replacement	5

item4	Qualitative	Scale of 1-8 importance to customer -	3
item5	Qualitative	Scale of 1-8 importance to customer - Options	4
item6	Qualitative	Scale of 1-8 importance to customer - Respectful	4
item7	Qualitative	Scale of 1-8 importance to customer - Courteous	3
item8	Qualitative	Scale of 1-8 importance to customer - Evidence of active	4

Part II: Data-Cleaning Plan

C1: Plan

1. **Create a Jupyter notebook in Google Colab. Use Python and the pandas' library to import data.**

```
import
pd.read
```

2. **Look at the shape, info, and head of the data to get an initial impression of the size and what is being examined. Compare against the provided data dictionary.**

```
.shape
.info()
.head()
```

3. **Drop unnecessary columns or duplicate columns.**

```
.drop(columns='column: 0')
```

4. **Rename columns**

```
.set_axis
```

5. **Reformat column data**

```
.astype
```

6. Look for duplicates

```
.duplicated()
.duplicated().value_counts()
```

7. Look for missing values

```
.isnull().sum()
sns.heatmap

dataframe.column.info()
.value_counts
plt.hist
.median()
```

8. Look for outliers

```
sns.boxplot
```

C2: Approach

- The plan began by first looking at the shape, information, and head of the data to get a broad overview of the data's contents. `.shape` will show the number of rows and columns in the dataset. `.info` will show the columns, non-null count, and data types present in the dataset. `.head` will show x amount of rows in their entirety. The plan then calls for examining each column and comparing it with the provided data dictionary for anomalies. The next step would be dropping columns, renaming columns, and reformatting data to address data quality issues. `.drop` to drop duplicate columns. `.set_axis` to change column names. `.astype` to reformat data types `.replace` and a dictionary was used to replace categorical data with numeric data. Then the steps would be to look for duplicates, missing values, and outliers. `.duplicated` and `duplicated().value_counts()` were used to search for duplicate values expressed as either True with present duplicates or False if there are no present duplicates. Missing values were examined using `.isnull().sum()` a heat map was also created using `sns.heatmap()` to provide a visualization to easily interpret the missing data. `plt.hist` was used to create histograms of data present for columns with missing data along with `.median` was used to get a full picture of data in columns when a large portion of data is missing. `sns.boxplot` was used to examine outliers for all quantitative variables. This data cleaning plan progresses from a broad examination to a focused approach, ensuring a clean dataset suitable for future analysis.

C3: Tools

- Python was chosen for data cleaning because of the experience I have in that language, its versatility in working with data, and the many libraries available for data-related tasks.

C4: Code

- See BethFuller_D206.ipynb for notebook or BethFuller_D206IPYNBPDF_02142024.pdf.

Part III: Data Cleaning (D1: Findings, D2: Justification, D3: Summary)

- **Finding:** CaseOrder and an Unnamed duplicate column were found with `.head()` and comparison with the data dictionary.
    - **Justification:** The additional column adds unnecessary data to the data frame.
    - **Summary:** `.drop` was used to remove and the updated `.head()` shows the column removed and a data frame that matches the data dictionary of 50 columns

- **Finding:** Nonstandard Python naming conventions were being used along with unclear column names in the data frame
    - **Justification:** By updating the column names to the standard Python style of lower_case_with_underscores (Python Software Foundation, n.d.) this change will allow for the easier calling of columns and future analysis. Column names were updated for clarity. For example, item1 became survey1_response which gives more insight in future data analysis.
    - **Summary:** `.set_axis` was used to update column names. The refreshed `.info()` shows a more legible and logical data frame.

- **Finding:** `.head()` shows nonstandard zip codes and `.info()` shows zip code being stored as an integer
    - **Justification:** To match standard mailing conventions the zip codes should be filled starting with 0's in addition zip code should be stored as a string as they are not numeric data. `.astype` and `.str.zfill(5)` were utilized to update zip codes.
    - **Summary:** `.head()` along with printing the row zip_code shows that the updated zip codes are all of the length 5 along with the appropriate leading zeros.

- **Finding:** Based on .nuiquie() and examination of columns with the data dictionary several columns can be changed from objects to columns to provide better analysis.
    - **Justification:** When a column contains a limited number of unique values categories are used to improve performance and perform operations faster. For the columns area, marital status, gender, contract, internet service, and payment method change from object to category.
    - **Summary:** This change can be examined in the .info, .unique of the data frame. It organizes our data better for future analysis and operations on the specified columns.

- **Finding:** In the time zone column they are listed as objects and there are too many unique values.
    - **Justification:** On the initial `.nunique` look at the data there are 25 timezones as objects. This seems high when the data seems to mainly come from the United States. Looking at this column with `.value_counts()` shows many individual cities could be better merged into more standard categories such as Atlantic, Eastern, Central, etc. (Wikipedia, "Time in the United States", 2024) to get a better view of where customers are located. `.astype` will be used to change timezone to a category and `.replace` will be used to merge different time zones into standard timezone categories.
    - **Summary:** After changing the data to category and merging the data from separate cities into standard time zones `.value_counts()` is now showing 8 timezones and where the bulk of the customers are located (eastern/central). This will be easier to utilize in analysis.

- **Finding:** When looking at `.info` and `.nunique` of the data education shows 12 different unique values this seems like a good column to be changed into an ordered categorical value since education typically follows an order - high school, bachelors, etc.
    - **Justification:** Upon looking at the initial .value_counts of the data the counts are all over. This would be better served as an ordered categorical using `pd.Categorical` which will allow the data to be stored from lowest amount of education to highest level of education. This will be accomplished using `pd.Categorical` and `ordered=True`.
    - **Summary:** After updating the data the data can be sorted and also a plot can be created in order of education. plot(kind='bar') was used with education level on the x axis and count on the y axis. This shows the most popular educational categories of our data in a visual and easier-to-view format.

- **Finding:** When looking at `.info` and `.nunique` of the data employment shows 5 different unique values this seems like again like a good column to be changed into an ordered categorical value since employment can also have a flow - unemployed to full-time employed.
    - **Justification:** As in education `.value_counts()` were scattered. The data is changed to a `pd.Categorical` and the categories are sorted from Unemployed to Full Time to analyze the data better.
    - **Summary:** After updating the employment data it was also easy to plot again to view with the x-axis being employment level and the y-axis being count. Ordering also allows easier sorting of the data.

- **Finding:** When using `.describe` to do an overview of the numerical data in the data frame population minimum being 0 stood out. Presumably, at least our customer lives in the town so the population would have to, at minimum, be 1. Using .sum to count the number of columns where population is 0 it was found that 97 of the columns were zero. Then by selecting the cities where the population is 0 in our data frame, it showed a long list of many different cities.
    - **Justification:** Since out of 10,0000 rows, this is approximately 1% of our data I have chosen to leave these values as 0. Google/research could be done to replace the particular missing values in the data frame but this seems unnecessary with the percentage of data missing in this particular column.
    - **Summary:** We know when doing future analysis that the population could be very slightly wrong or skewed but with this small percentage of missing data leaving it alone seems better than employing a median, mean, or some other value into the cities since looking at the cities missing values they are all over the place in terms of size.

- **Finding:** When searching for duplicates with `.duplicated()` and `.value_counts()` there were no duplicates found.
    - **Justification:** Nothing needs to be done since no duplicates were found.
    - **Summary:** No duplicates were present but they were searched for so we know we are working with a data frame that does not contain duplicates.

- **Finding:** By using `.isnull().sum()` several columns were found to contain missing values a histogram was created with `sns.headmap` to further investigate the columns containing missing values. children, age, income, techie, phone_service, tech_support,

tenure and bandwidth_gb_year contain missing values ranging up to 25% of missing values so investing these columns further is required.
    - **Justification:** use `.info()` and `.value_counts()` to further investigate columns with missing data. This is a relatively high percentage of data missing there is valuable data in other rows so we do not want to simply drop these columns and lose out on the other data in these rows. We will need to replace the values in these rows with values that make sense for the columns.
    - **Summary:** Children and age will be dealt with using random values. Income, tenure, and bandwidth gb per year will be dealt with using median values. Techie, phone, and tech support will be dealt with using random numbers with the same percentages as the present data.

- **Finding:** Children are missing 2,495 values.
    - **Justification:** By looking at the histogram for children and also the median (1) it is showing that most people either have 0 or 1 children. Using this logic to randomly assign either 0 or 1 child to the missing customers.
    - **Summary:** By using this approach we maintain our data we maintain a very similar shape histogram of children's data and the same median.

- **Finding:** Age is missing 2,475 values.
    - **Justification:** Looking at our histogram and also the `.describe()` of our age data it shows customers between 18-89 and pretty evenly spread throughout. We will use random to randomly assign ages between 18 and 89 to the missing customers.
    - **Summary:** Using `.describe()` and also checking our histogram the data remains very similar to the prior data.

- **Finding:** Techie, phone, and tech support were all Yes and No questions.
  Using .value_counts(normalize=1) we can see the percentages our data was giving.
    - **Justification:** By using .np.random.choice with the same percentage of Yes and No's from the data to replace the missing values.
    - **Summary:** By using the same percentage of Yes / No from these questions the integrity of the answers that were not null is maintained in the updated data frame.

- **Finding:** To search for outliers boxplots were utilized on all numeric columns along with `.describe()`, `.sum()`, and `value_count()` where needed. Outliers were found in population, children, income, outage_sec_perweek, email, contacts, yearly_equip_failure, and monthly_charge.
    - **Justification:** While outliers can be bad data points and can alter data when doing statistical analysis many of our outliers are true outliers and should be left. We will investigate each of these columns with outliers to determine what should be done.
        - **Population:** This box plot along with our further investigation into the columns still shows 97 customers with missing population values. Since the cities were so varied in size for the time being the data will be kept at 0 instead of replacing with a random value or a median.
        - **Children:** showing outliers where people have more than 7 children. This is possible and makes sense and should be left because while these are outliers it is still accurate data.

- **Income:** Incomes can be above 250,000 which is where our highest outliers lie. These were investigated and while they may slightly alter our mean they seem accurate and are going to be kept.
- **Outage sec per week:** There are many outliers above 30 seconds per week which are valid and should be important to the business. The outliers in the outage second per week that are showing below 0 however do not make sense. There can be 0 outage seconds per week but below that does not make sense. There are 11 columns where this is the case. Instead of removing the data from these rows entirely we will change these values to 0.
- **Email:** Outliers in the email section make sense. The minimum is 1 the maximum being 23. These are not outlandish numbers depending on the individual customers' needs.
- **Contacts:** Contacts also make sense being from 0 contacts to a maximum of 7. The data will be left alone here.
- **Yearly equipment failure:** Data seems accurate for yearly equipment failure even though it again contains outliers. The data stretches from 0 failures to 6. This makes sense and should be left.
- Monthly charge: The minimum of monthly charge starts at $77.51 and the maximum is $315.88. While there are a few of outliers in our box plot further investigation shows only 3 above $300 and without understanding the bill further this seems plausible and the data will be kept.

- **Summary:** The only column that was acted upon with outlier data was outage_sec_perweek since there is no possible way to have a negative value in this column these 11 rows were replaced with 0's. The other column that has issues with 1% of its data is the population column. This column will be left for the time being due to the varied nature of the city sizes missing data.

D4: Mitigation Code

- See BethFuller_D206.ipynb for notebook or BethFuller_D206IPYNBPDF_02142024.pdf.

D5: Clean Data

- See churndf_clean.csv

D6: Limitations

- This data set, like many likely to be encountered in the real world, contained many issues. The biggest of which is that all of the data is likely to be generated at random creating a nonsense data frame. Through casual observation when inspecting the data frame many things did not add up but to confirm this was put to a quick test. First by comparing Doctorate degrees with age, second by age and retirement status and third city and population. There were many young Doctorates, and retirees and when sorted by size many cities came above New York which has an approximate population size of 8 million people and was listed at just 436,270 in the data frame (Wikipedia, "New York City", 2024). Exploring insights into this small dataset could be very unreliable and inaccurate. In addition to the random nature of the dataset, it contained more standard issues such as missing data (up to 25% in some columns), outliers, naming conventions not being followed, and other formatting issues as addressed in detail.

D7: Impact of Limitations

- Due to the limitations discussed above and mostly due to the random nature of the dataset, further efforts in data collection may be necessary to ensure more reliable insights. Without

additional data, it would be best to exercise caution and refrain from drawing serious conclusions or taking significant action based solely on the current dataset.

E1: Principal Components

- population
- children
- age
- income
- outage_sec_perweek
- email
- contacts
- yearly_equip_failure
- techie

E2: Criteria Used

- The principal components of the data set are found after normalizing the data and selecting the components with an eigenvalue above 1. An eigenvalue above 1 suggests higher importance, less than one is considered trivial (Larose & Larose 2019).

E3: Benefits

- Principal component analysis is a popular data transformation technique because it can provide insight into the most important aspects of a given data set. Following a PCA focus can be put on the principal components to make the most efficient analysis and model building.

F. Video

- https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=606205bc-632a-481f-802f-b116012e7c47

Part IV: Supporting Documents

G: Sources of Third-Party Code

- Courseware Lesson 7: How to Perform PCA in Python (Larose & Larose 2019)

H. Sources

- Larose, C. D., & Larose, D. T. (2019). Data science using Python and R. ISBN-13: 978-1-119-52684-1.
- Python Software Foundation. (n.d.). PEP 8 -- Style Guide for Python Code. Python Enhancement Proposals. Retrieved February 12, 2024, from https://peps.python.org/pep-0008/
- Tableau. (n.d.). Box-and-Whisker Plot. Tableau Reference Library. Retrieved from https://www.tableau.com/data-insights/reference-library/visual-analytics/charts/box-whisker
- Wikipedia contributors. (2024, February 12). New York City. In Wikipedia, The Free Encyclopedia. Retrieved February 12, 2024, from https://en.wikipedia.org/wiki/New_York_City
- Wikipedia contributors. (2024, February 12). Time in the United States. In Wikipedia, The Free Encyclopedia. Retrieved February 12, 2024, from https://en.wikipedia.org/wiki/Time_in_the_United_States

I: Professional Communication