Performance Assessment:

Data Cleaning

Ryan James Calabio

D206: Data Cleaning

Mar 25, 2024

A: Research Question

The research question I have for the churn dataset is “What customer specific factors affect churn?”. This question can be addressed as it has a churn column, and a lot of data that is relevant to customers. Information like location data, job, age, children, education, employment, and gender can all potentially play a role in whether the churn value is yes or no. All the variables in the dataset can be used in some regard to help either categorize a customer by id, or potentially identify a relationship to the churn value. This is also a relevant question to the business, as it would make sense to understand what affects churn and then optimize service to account for those factors to reduce it.

B: Required Variables

Variable Name	Data Type	Description/Definition	Example Value
CaseOrder	int64	Keeps the order of data	1
Customer_id	object	Identifies the customer	K409198
Interaction	object	ID that are relevant for customer specific actions	aa90260b-4141-4a24-8e36-b04ce1f4f77b
City	object	City from billing statement for customer	Point Baker
State	object	State from billing statement for customer	AK
County	object	County from billing statement for customer	Yamhill
Zip	int64	Zip from billing statement for customer	92014
Lat	float64	GPS from billing statement for customer	33.58016

Lng	float64	GPS from billing statement for customer	-85.13241
Population	int64	Pop. from census data nearby	13863
Area	object	Type from census	Suburban
Timezone	object	Timezone of location	America/Los_Angeles
Job	object	Reported occupation	Solicitor
Children	float64	Number of children in customer's household	1
Age	float64	Age of the customer	48
Education	object	Highest degree of the customer	Doctorate Degree
Employment	object	Whether they are employed or not	Retired
Income	float64	Annual income reported by customer	18925.23
Marital	object	Marital status reported by customer	Separated
Gender	object	Self identification of gender	Female
Churn	object	If customer canceled service	No
Outage_sec_perweek	float64	Avg seconds system was out in Neighborhoods	7.110666
Email	int64	How many emails sent to customer past year	14
Contacts	int64	Tech support contact count	1
Yearly_equip_failure	int64	Number of times equipment failed and was replaced for the last year	0
Techie	object	If customer self identifies as technical	Yes
Contract	object	Contract term for customer	One year

Port_modem	object	If customer has portable modem	Yes
Tablet	object	If customer has tablet	Yes
InternetService	object	If customer owns internet service	Fiber Optic
Phone	object	If customer has a phone service	Yes
Multiple	object	If customer has multiple lines	Yes
OnlineSecurity	object	If customer has online security	Yes
OnlineBackup	object	If customer has online backup	Yes
DeviceProtection	object	If customer has device protection	Yes
TechSupport	object	If customer has technical support	Yes
StreamingTV	object	If customer hat tv streaming	Yes
StreamingMovies	object	If customer has movie streaming	Yes
PaperlessBilling	object	If customer has paperless billing	Yes
PaymentMethod	object	The payment method used to pay for bill	Credit Card (automatic)
Tenure	float64	Months customer has been with provider	12.80616
MonthlyCharge	float64	Amount charged on bill per month	154.0171
Bandwidth_GB_Year	float64	Average amount of data used. In gb and per year	713.0633
item1	int64	Timely response rating from 1 to 8	1
item2	int64	Timely fixes rating from 1 to 8	2
item3	int64	Timely replacements rating from 1 to 8	3

item4	int64	Reliability rating from 1 to 8	4
item5	int64	Options rating from 1 to 8	5
item6	int64	Respectful response rating from 1 to 8	6
item7	int64	Courteous exchange rating from 1 to 8	7
item8	int64	Evidence of active listening rating from 1 to 8	8

C1: Plan To Assess Quality of Data

Through the data cleaning process for this project, there were various techniques used through the code. Many of these techniques were used to detect problems like duplicates, missing values, outliers, and re-expression of categorical variables. To determine if there were duplicate values, I used functions to show me what columns had duplicate values. Using the .duplicated() function and the .duplicated().sum() combination fo functions I was able to determine that there were not duplicate values in the data. To determine if there were missing values, I also used a mixture of various packages. To start, I used .isnull().sum() to see what columns had missing values. This showed me that Children, Age, Income, Techie, InternetService, Phone, TechSupport, Tenure, and Bandwidth_GB_Year all had null values contained within their respective columns. I visualized this further using a missingno matrix and matplotlib. To detect outliers, I calculated the z-scores for columns and created histograms from the created z-score columns. I chcecked for outliers in quantitative variables which included: population, children, income,

outage_sec_perweek, email, contacts, yearly_equip_failure, tenure, MonthlyChargge, and Bandwidth_GB_year.

C2: Justification of Approach

For duplicates, I used the .duplicated().sum() function because it allowed me to see the actual number of nulls. By returning a result that specified no nulls, I could know that this was not a problem I had to fix with the data.

For missing values, using the .isnull().sum() functions allowed me to see exactly what functions need to be dealt with. With missingno, we can see the severity of the missing values and with matplotlib we can create histograms to determine what the distribution of each column is.

For outliers, using scipy to calculate s-scores allows us to keep scale consistent to a level where we can justify that a certain value is an outlier. Matplotlib lets us create visuals that take this further by showing which have severe outliers.

C3: Justification of Tools

For this performance assessment, I used Python as it is the coding language that I am most familiar with. I use it at work and thus wanted to focus on it as an area of improvement and growth. Throughout the project, I used these packages to go through the data cleaning process: numpy, pandas, seaborn, missingno, matplotlib, scipy, and sklearn. I used numpy to

I used pandas to manipulate the data frames within the notebook. Next, seaborn, missingno, and matplotlib was used to create visualizations within the notebook. Then I used scipy to calculate

z-scores to assist in determining what variables had outliers. Finally, I used sklearn to do principal component analysis.

C4: Provide The Code

See code attatched

D1: Cleaning Findings

While cleaning the churn data, I used the .duplicated() function to find that there were no duplicates present within the data. However, I did find that there were a few variables with missing values present. Children, Age, Income, Techie, InternetService, Phone, TechSupport, Tenure, and Bandwidth_GB_Year were all variables that had missing values. Children had 2495 missing values. Age has 2475 missing values. Income had 2490 missing values. Techie had 2477 missing values. InternetService has 2129 missing values. Phone has 1026 missing values. TechSupport has 991 missing values. Tenure has 931 missing values. Bandwidrth_GB_Year has 1021 missing values.

For outliers, I used histograms to see the z-scores of each quantitative metric. Two standard deviations are a z-score of two, so anything over two and under negative two I will consider an outlier in this situation. Every variable except tenure has a value with a z-score greater than two or less than negative 2.

D2: Justification of Mitigation Methods

Since there were no duplicates present within the churn dataset, no process needed to be done to fix duplicates.

For the nulls, I start by finding the distribution of all the quantitative variables using matplotlib. Children was skewed right, Age was uniform, Income was skewed right, Tenure was bi-modal, and Bandwidth_GB_year was bi-modal. Because of the distributions, the method to replace the nulls was replace Children with the median, replace Income nulls with median, replace Tenure nulls with mode, and replace Bandwidth_GB_year nulls with mode. For the categorical data, I

used the mode regardless since the data does not have a mean or a median if its categorical. This includes Techie, InternetService, Phone, and TechSupport. For the float data I used mean because no mode would exist for that data. This includes Bandwidth_GB_Year and Tenure.

For the outliers, I used imputation to adjust values in the columns with outliers which would be z-scores greater than two or less than negative two. Depending on the distribution of the z-scores, I applied a different type of imputation. For normal distribution, we use the means to replace the outliers. For skewed right or left, we use the median to replace the values. For bimodal distributions, we replace outliers with mode.

D3: Summary of The Outcomes

Throughout this process of data cleaning, I have done a multitude of tasks in order to clean the data. This began with understanding all the columns' data types and what problems there were with each variable. This included detecting nulls, outliers, and duplicates. Nulls and outliers were discovered using functions like .duplicated() and .isnull(). Using the df.info() function I determine that there are no nulls remaining in the data. Next is to check the histograms to verify if there are still significant outliers remaining. The variables I removed outliers for were: population, children, income, outage_sec_perweek, email, contacts, yearly_equip_failure, tenure, MonthlyChargge, and Bandwidth_GB_year. Looking at the histograms comparing the original dataset and the new dataset, many of the variables have removed a lot of the excessively large or low numbers.

D4: Mitigation Code

See code attached.

D5: Clean Data

See CSV file attached.

D6: Limitations

There are a few disadvantages to the methods that I utilized in cleaning the data. For filling in categorical data with the mode, it may overemphasize the frequency of the mode by too much within the data. Also, for using mean only for float data when it is a bimodal distribution, it is not technically following the ideal imputation format. However, mode does not exist. Dropping these values does not make sense because it is not a significantly large amount enough of nulls to do such a drastic thing to the dataset. Since no duplicates existed, nothing had to be treated within the data. Finally, for dealing with outliers, since outliers were done after treating nulls, the distribution may once again change from the method used to treat the nulls, changing the ideal imputation method.

D7: Impact of Limitations

The limitations of how the data was cleaned may affect how an analyst receives an answer to the research question. The question that I asked was what factors affect whether a customer will be true or false in the churn column. Limitations like how the categorical data was imputed using the mode may mean that the impact of a certain categorical variable is overemphasized in the impact it has on the churn variable in relation to a customer.

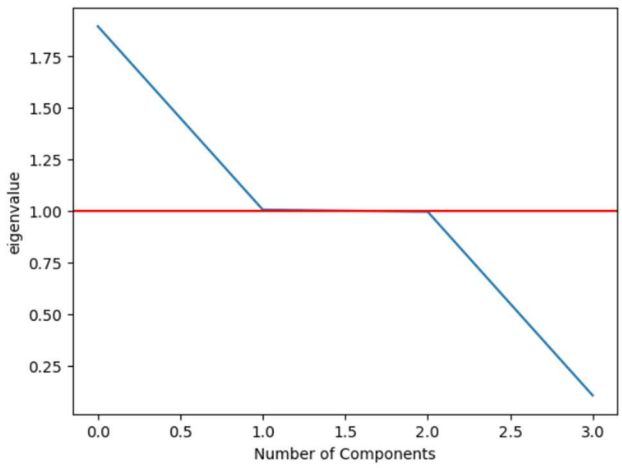
E1: Principal Components

For principal component analysis, the best variables to use are quantitative and continuous variables. Among the dataset, the quantitative and continuous variables included Income, Tenure, MonthlyCharge, and Bandwidth_GB_Year. These are the variables I decided to use for my principal component analysis. The loading matrix is as follows:

	PC1	PC2	PC3	PC4
Income	0.007122	0.680951	0.732294	-0.000238
Tenure	0.706112	0.037428	-0.041901	-0.705867
MonthlyCharge	0.034862	-0.731326	0.679697	-0.044252
Bandwidth_GB_Year	0.707205	-0.008178	0.000955	0.706960

E2: Criteria Used

We can determine which PCs should be kept based on the scree plot. The scree plot looks like:



Also, according to the Kaiser rule: PCs with an eigenvalue >= 1 should be kept. In this graph, 0, 1, and 2, represent PCs that should be kept since 3 is below the red line indicated an eigenvalue of 1. This means that PC1, PC2, and PC3 should be retained.

E3: Benefits

The principal component analysis allows us to understand the relationships between our variables and how they may affect churn. Based on how we identify and interpret these relationships, we can advise business decisions and implement changes that may decrease churn. For instance, our principal component one helps us identify that there is a positive relationship between tenure and bandwidth_gb_year. This means that if we want to lower churn and increase tenure, focusing on factors that increase bandwidth_gb_year may help us with that. Principal component two allows us to see the negative relationship between income and monthly charge. This may be because customers that have higher income have more experience managing money, and might minimize their expenses for unnecessary services. Principal component three indicates that there also may be a positive relationship between income and MonthlyCharge. This may be because as income increases, the more expensive products and services the customer may opt for. If we want to decrease churn, we can target ideal products based on income so that a customer doesn't overextend the amount of money they can spend and cancel. The fourth principal component analysis indicates that it should be dropped by our scree plot and Kaiser rule.

G: Sources of Third-Party Code

No third-party code references were used

H: Sources

Larose, Chantal D, and Daniel T Larose. *Data Science Using Python and R*. Hoboken, Wiley, 2019.

Browne-Anderson, Hugo, et al. "ORGANIZATION TRACK D206 - Data Cleaning." *DataCamp*, DataCamp, app.datacamp.com/learn/custom-tracks/custom-d206-data-cleaning. Accessed 22 Mar. 2024.

"Msno.matrix() Shows an Error When I Use Any Venv Using Pyenv." *Stack Overflow*, 2023, stackoverflow.com/questions/75525029/msno-matrix-shows-an-error-when-i-use-any-venv-using-pyenv. Accessed 22 Mar. 2024.