D208 MULTIPLE REGRESSION PREDICTIVE MODELING [TASK 1]

Performance Assessment Task

WGU - MSDA

Multiple Regression Predictive Modeling using Cleaned Churn Dataset

Richard Flores
Rflo147@wgu.edu
Student ID: 006771163

Table of Contents

Part I: Research Question

A1. Research Question

As the world becomes ever more Data Driven, users are increasingly using their mobile and telecom devices for commercial productivity such as high-resolution video chats and for streaming entertainment in the form of music and movies from platforms like Netflix and YouTube.

The question answered in this report is, <u>can Predictive Modeling using Multiple Regression predict a customer's annual data use based on provided variables?</u>

A2. Data Analysis Objectives

Valuable insight can be gained from an analysis of a customer's data usage extracted from the provided telecommunications churn dataset. As the need for data consumption grows, telecom companies can greatly benefit from predicting data used by each customer and an overall prediction of data use in the future to prepare network infrastructure. These insights can also help with provisioning and pricing customer's data plans and assist the marketing team with developing attractive incentives based on a customer's data need.

Part II: Method Justification

B1. Multiple Regression Model Assumptions

The assumptions of the regression models are as follows:

First, multiple regression requires that the relationship between the independent variable and the dependent variable is linear. The linearity assumption is best tested on a distribution graph.

Secondly, multiple linear regression analysis requires the error between the observed value and the predicted value to be normally distributed (ie, regression residual). This hypothesis can be confirmed by studying histograms or Q-Q charts.

Third, multiple regression assumes that the data are not multiple collinear. When individual variables are excessively dependent on each other, there will be many linearities (Statistics Solutions 2021).

B2. Benefits of Tools (Python)

As per previous assessment submissions, I will continue to use the Python programming language and the PyCharm IDE to develop and test code. For ease of displaying code in a structured and organized manner I will refactor the code in Jupyter Notebook. The benefit of writing code in Python is the language is cross-platform being readily available on Windows, Mac, and Linux increasing access to a broad range of developers. The Python language includes a large community of Data Science developers contributing libraries that are essential to this assessment such as Pandas, NumPy, and SciPy. While Python is similar in comparison to R, Python offers advantages in Speed, Data Visualization, and Interoperability (Larose 2019).

B3. Explanation of Multiple Regression Technique

Multiple regression is a good way to analyze the research questions and provide useful information for the telecom company. The variable to predict is the actual amount of GB per year which is a continuous variable (data size). In addition, some explanatory variables included in the dataset expand our understanding of how to try to predict data usage in the future i.e., age, children, income work. When adding and subtracting independent variables we will determine if the regression equation is positive or negative in relation with target variables and how they affect company profits (Li, L 2019).

Part III: Data Preparation

C1. Data Preparation Goals

For this assessment, I will begin with the cleaned 'churn_clean' dataset previously prepared in a prior course. In order to prepare the selected dataset for Multiple Regression analysis the following goals and manipulations will need to be accomplished.

Data will need to be imported and formatted for regression using necessary libraries such as NumPy, Pandas, Seaborn, Matplotlib, PyLab, and Sklearn. In order to provide meaningful insight that is clear to stakeholders the customer survey columns must be renamed from 'item' to a variable description that accurately reflects the survey response.

With 50 features/columns of data, we will need to remove columns not relevant to the Regression Analysis. Also, to ensure data is accurate we will need to verify no missing datapoints exist in the dataset.

It is important to provide data that can be used by the Python libraries therefor we must remap variables from 'Yes/No' into '1 or 0' with dummy variables that can be calculated numerically. After creating the dummy variables we will then need to remove original 'Yes/No' columns to prevent redundancy in the dataset. Finally the target Bandwidth_GB_Year feature needs to be moved to end of dataset.

For this analysis, the target continuous variable is "Bandwidth_GB_Year". The goal is to train and test the model on the dataset to provide insight onto the amount of data a customer will annually consume. In completing the analysis, it is possible to uncover relevant continuous predictor variables as well as categorical predictor variables.

For the customer survey responses which represent discrete ordinal predictors, the data will be manipulated for clarity with the following changes:

- Item1: Timely response
- Item2: Timely fix
- Item3: Timely replacement
- Item4: Reliability
- Item5: Options
- Item6: Respectful
- Item7: Courteous
- Item8: Active listening

The surveys are calculated on a score range from 1 to 8, with 1 representing the most important factor and 8 the least important.

C2. Summary Statistics

The original churn dataset contains 50 features/columns and 10,000 records/rows. As noted in preparation, irrelevant features have been removed from the dataset which are:

"Area, CaseOrder, City, County, Customer_id, Interaction, Job, Lat, Lng, Marital, PaymentMethod, Population, State, TimeZone, UID, Zip"

As also stated in preparation, binomial data with entries such as "Yes/No" were refactored as "1/0".

From the initial 50 features, 34 relevant columns remain to support analysis of the target variable. The remaining dataset is free from null, NaN, and missing data point values and no outliers remain.

Histograms and boxplots which were used to measure central tendency show normal distributions for features "email, monthly_charge, and outage_sec_perweek".

A histogram generated for "bandwidth_GB_year" and "tenure" show bimodal distributions correlating to a linear relationship.

From the data provided, we can determine the average telecom customer is 53 years old placed in a standard deviation of 20 years, has two children with a standard deviation of 2, an income of $40,000 with standard deviation of $30,000, experienced 10 outage seconds per week, received 12 marketing emails, contacted technical support once or less, had one or less equiptment failures a year, has a tenure of 34.5 months, and uses 3,400 GB's per year.

C3. Steps used to Prepare Data for Analysis

- Import the 'clean_churn' dataset into a Pandas dataframe for analysis.
- Rename features in the survey responses to better describe the items.
- Describe the various features and data to prepare relevant items.
- Create a view of the summary statistics.
- After review, remove features that are not relevant to analyzing the target variable.
- Review record data to check for anomalies, outliers, missing data and other data that could become obstacles in the analysis.
- Utilize dummy variables in order to numerically analyze data by changing "Yes/No" responses to binary "1/0" responses.
- Create necessary univariate and bivariate visualizations.
- Move the target variable "bandwidth_GB_year" as the final feature.
- Review manipulated data and export the new dataframe to CSV for review.

```python
# Standard library imports, and Visualization, Statistics, SciKit, ChiSquare libraries
import numpy as np
import pandas as pd
from pandas import Series, DataFrame

import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

import pylab
from pylab import rcParams

import statsmodels.api as sm
import statistics
from scipy import stats

import sklearn
from sklearn import preprocessing
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import classification_report

from scipy.stats import chisquare
from scipy.stats import chi2_contingency

#Skip warning messages
import warnings
warnings.filterwarnings('ignore')

import matplotlib as mpl
COLOR = 'white'
mpl.rcParams['text.color'] = COLOR
mpl.rcParams['axes.labelcolor'] = COLOR
mpl.rcParams['xtick.color'] = COLOR
mpl.rcParams['ytick.color'] = COLOR
```

```python
 # Load churn dataset into a Pandas dataframe
churn_df = pd.read_csv('/Users/Richard/OneDrive - Western Governors University/MSDA/D208/Databases/churn/churn_clean.csv')

# Rename 8 customer survey features to represent descriptions for clarity
churn_df.rename(columns = {'Item1':'TimelyResponse',
'Item2':'Fixes',
'Item3':'Replacements',
'Item4':'Reliability',
'Item5':'Options',
'Item6':'Respectfulness',
'Item7':'Courteous',
'Item8':'Listening'},
inplace=True)
```

(DataCamp 2021).

```
# Verify new dataframe is active and correct
churn_df
```

	CaseOrder	Customer_id	Interaction	UID	City	State	County	Zip	Lat	Lng	...	MonthlyCharg
0	1	K409198	aa90260b-4141-4a24-8e36-b04ce1f4f77b	e885b299883d4f9fb18e39c75155d990	Point Baker	AK	Prince of Wales-Hyder	99927	56.25100	-133.37571	...	172.4555
1	2	S120509	fb76459f-c047-4a9d-8af9-e0f7d4ac2524	f2de8bef964785f41a2959829830fb8a	West Branch	MI	Ogemaw	48661	44.32893	-84.24080	...	242.6325
2	3	K191035	344d114c-3736-4be5-98f7-c72c281e2d35	f1784cfa9f6d92ae816197eb175d3c71	Yamhill	OR	Yamhill	97148	45.35589	-123.24657	...	159.9475
3	4	D90850	abfa2b40-2d43-4994-b15a-989b8c79e311	dc8a365077241bb5cd5ccd305136b05e	Del Mar	CA	San Diego	92014	32.96687	-117.24798	...	119.9568
4	5	K662701	68a861fd-0d20-4e51-a587-8a90407ee574	aabb64a116e83fdc4befc1fbab1663f9	Needville	TX	Fort Bend	77461	29.38012	-95.80673	...	149.9483
...	...	...	...	...	...	...	...	...	...	...	...	
9995	9996	M324793	45deb5a2-ae04-4518-bf0b-c82db8dbe4a4	9499fb4de537af195d16d046b79fd20a	Mount Holly	VT	Rutland	5758	43.43391	-72.78734	...	159.9794
9996	9997	D861732	6e96b921-0c09-4993-bbda-a1ac6411061a	c09a841117fa81b5c8e19afec2760104	Clarksville	TN	Montgomery	37042	36.56907	-87.41694	...	207.4811
9997	9998	I243405	e8307ddf-9a01-4fff-bc59-4742e03fd24f	9c41f212d1e04dca84445019bbc9b41c	Mobeetie	TX	Wheeler	79061	35.52039	-100.44180	...	169.9741
9998	9999	I641617	3775ccfc-0052-4107-81ae-9657f81ecdf3	3e1f269b40c235a1038863ecf6b7a0df	Carrollton	GA	Carroll	30117	33.58016	-85.13241	...	252.6240
9999	10000	T38070	9de5fb6e-bd33-4995-aec8-f01d0172a499	0ea683a03a3cd544aefe8388aab16176	Clarkesville	GA	Habersham	30523	34.70783	-83.53648	...	217.4840

10000 rows × 50 columns

```
# Describe Churn dataset
churn_df.describe()
```

	CaseOrder	Zip	Lat	Lng	Population	Children	Age	Income	Outage_sec_perweek	Email	...
count	10000.00000	10000.000000	10000.000000	10000.000000	10000.000000	10000.0000	10000.000000	10000.000000	10000.000000	10000.000000	...
mean	5000.50000	49153.319600	38.757567	-90.782536	9756.562400	2.0877	53.078400	39806.926771	10.001848	12.016000	...
std	2886.89568	27532.196108	5.437389	15.156142	14432.698671	2.1472	20.698882	28199.916702	2.976019	3.025898	...
min	1.00000	601.000000	17.966120	-171.688150	0.000000	0.0000	18.000000	348.670000	0.099747	1.000000	...
25%	2500.75000	26292.500000	35.341828	-97.082812	738.000000	0.0000	35.000000	19224.717500	8.018214	10.000000	...
50%	5000.50000	48869.500000	39.395800	-87.918800	2910.500000	1.0000	53.000000	33170.605000	10.018560	12.000000	...
75%	7500.25000	71866.500000	42.106908	-80.088745	13168.000000	3.0000	71.000000	53246.170000	11.969485	14.000000	...
max	10000.00000	99929.000000	70.640660	-65.667850	111850.000000	10.0000	89.000000	258900.700000	21.207230	23.000000	...

8 rows × 23 columns

(DataCamp 2021).

```python
# Remove features not relevant to the proposed analysis question
churn_df = churn_df.drop(columns=['CaseOrder', 'Customer_id', 'Interaction', 'UID', 'City', 'State', 'County', 'Zip', 'Lat',
                                   'Lng', 'Population', 'Area', 'TimeZone', 'Job', 'Marital', 'PaymentMethod'])
churn_df.describe()
```

	Children	Age	Income	Outage_sec_perweek	Email	Contacts	Yearly_equip_failure	Tenure	MonthlyCharge	Bandwid
count	10000.0000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	1(
mean	2.0877	53.078400	39806.926771	10.001848	12.016000	0.994200	0.398000	34.526188	172.624816	[illegible]
std	2.1472	20.698882	28199.916702	2.976019	3.025898	0.988466	0.635953	26.443063	42.943094	[illegible]
min	0.0000	18.000000	348.670000	0.099747	1.000000	0.000000	0.000000	1.000259	79.978860	
25%	0.0000	35.000000	19224.717500	8.018214	10.000000	0.000000	0.000000	7.917694	139.979239	[illegible]
50%	1.0000	53.000000	33170.605000	10.018560	12.000000	1.000000	0.000000	35.430507	167.484700	[illegible]
75%	3.0000	71.000000	53246.170000	11.969485	14.000000	2.000000	1.000000	61.479795	200.734725	[illegible]
max	10.0000	89.000000	258900.700000	21.207230	23.000000	7.000000	6.000000	71.999280	290.160419	[illegible]

```python
# Verify there are no missing data points in the dataframe
data_nulls = churn_df.isnull().sum()
print(data_nulls)
```

```
Children                0
Age                     0
Income                  0
Gender                  0
Churn                   0
Outage_sec_perweek      0
Email                   0
Contacts                0
Yearly_equip_failure    0
Techie                  0
Contract                0
Port_modem              0
Tablet                  0
InternetService         0
Phone                   0
Multiple                0
OnlineSecurity          0
OnlineBackup            0
DeviceProtection        0
TechSupport             0
StreamingTV             0
StreamingMovies         0
PaperlessBilling        0
Tenure                  0
MonthlyCharge           0
Bandwidth_GB_Year       0
TimelyResponse          0
Fixes                   0
Replacements            0
Reliability             0
Options                 0
Respectfulness          0
Courteous               0
Listening               0
dtype: int64
```

(DataCamp 2021).

```python
# Convert all "Yes/No" data into binary "1/0" representation
churn_df['DummyChurn'] = [1 if v == 'Yes' else 0 for v in churn_df['Churn']]
churn_df['DummyContract'] = [1 if v == 'Two Year' else 0 for v in churn_df['Contract']]
churn_df['DummyDeviceProtection'] = [1 if v == 'Yes' else 0 for v in churn_df['DeviceProtection']]
churn_df['DummyGender'] = [1 if v == 'Male' else 0 for v in churn_df['Gender']]
churn_df['DummyInternetService'] = [1 if v == 'Fiber Optic' else 0 for v in churn_df['InternetService']]
churn_df['DummyMultiple'] = [1 if v == 'Yes' else 0 for v in churn_df['Multiple']]
churn_df['DummyOnlineBackup'] = [1 if v == 'Yes' else 0 for v in churn_df['OnlineBackup']]
churn_df['DummyOnlineSecurity'] = [1 if v == 'Yes' else 0 for v in churn_df['OnlineSecurity']]
churn_df['DummyPaperlessBilling'] = [1 if v == 'Yes' else 0 for v in churn_df['PaperlessBilling']]
churn_df['DummyPhone'] = [1 if v == 'Yes' else 0 for v in churn_df['Phone']]
churn_df['DummyPort_modem'] = [1 if v == 'Yes' else 0 for v in churn_df['Port_modem']]
churn_df['DummyStreamingTV'] = [1 if v == 'Yes' else 0 for v in churn_df['StreamingTV']]
churn_df['DummyTablet'] = [1 if v == 'Yes' else 0 for v in churn_df['Tablet']]
churn_df['DummyTechSupport'] = [1 if v == 'Yes' else 0 for v in churn_df['TechSupport']]
churn_df['DummyTechie'] = [1 if v == 'Yes' else 0 for v in churn_df['Techie']]
churn_df['StreamingMovies'] = [1 if v == 'Yes' else 0 for v in churn_df['StreamingMovies']]
```

```python
# Drop original categorical features from dataframe
churn_df = churn_df.drop(columns=['Churn',
                                  'Contract',
                                  'DeviceProtection',
                                  'Gender',
                                  'InternetService',
                                  'Multiple',
                                  'OnlineBackup',
                                  'OnlineSecurity',
                                  'PaperlessBilling',
                                  'Phone','Port_modem',
                                  'StreamingMovies',
                                  'StreamingTV',
                                  'Tablet',
                                  'TechSupport',
                                  'Techie'])
churn_df.describe()
```

	Children	Age	Income	Outage_sec_perweek	Email	Contacts	Yearly_equip_failure	Tenure	MonthlyCharge	Bandwid
count	10000.0000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	[illegible]
mean	2.0877	53.078400	39806.926771	10.001848	12.016000	0.994200	0.398000	34.526188	172.624816	[illegible]
std	2.1472	20.698882	28199.916702	2.976019	3.025898	0.988466	0.635953	26.443063	42.943094	[illegible]
min	0.0000	18.000000	348.670000	0.099747	1.000000	0.000000	0.000000	1.000259	79.978860	
25%	0.0000	35.000000	19224.717500	8.018214	10.000000	0.000000	0.000000	7.917694	139.979239	[illegible]
50%	1.0000	53.000000	33170.605000	10.018560	12.000000	1.000000	0.000000	35.430507	167.484700	[illegible]
75%	3.0000	71.000000	53246.170000	11.969485	14.000000	2.000000	1.000000	61.479795	200.734725	[illegible]
max	10.0000	89.000000	258900.700000	21.207230	23.000000	7.000000	6.000000	71.999280	290.160419	[illegible]

8 rows × 33 columns

```python
df = churn_df.columns
print(df)
```

```
Index(['Children', 'Age', 'Income', 'Outage_sec_perweek', 'Email', 'Contacts',
       'Yearly_equip_failure', 'Tenure', 'MonthlyCharge', 'Bandwidth_GB_Year',
       'TimelyResponse', 'Fixes', 'Replacements', 'Reliability', 'Options',
       'Respectfulness', 'Courteous', 'Listening', 'DummyChurn',
       'DummyContract', 'DummyDeviceProtection', 'DummyGender',
       'DummyInternetService', 'DummyMultiple', 'DummyOnlineBackup',
       'DummyOnlineSecurity', 'DummyPaperlessBilling', 'DummyPhone',
       'DummyPort_modem', 'DummyStreamingTV', 'DummyTablet',
       'DummyTechSupport', 'DummyTechie'],
      dtype='object')
```

(DataCamp 2021).

```python
# Move Bandwidth_GB_Year to end of dataset as target
churn_df = churn_df[['Children',
                     'Age', 'Income', 'Outage_sec_perweek', 'Email',
                     'Contacts', 'Yearly_equip_failure', 'Tenure',
                     'MonthlyCharge', 'TimelyResponse', 'Fixes',
                     'Replacements', 'Reliability', 'Options',
                     'Respectfulness', 'Courteous', 'Listening',
                     'DummyGender', 'DummyChurn', 'DummyTechie',
                     'DummyContract', 'DummyPort_modem', 'DummyTablet',
                     'DummyInternetService', 'DummyPhone', 'DummyMultiple',
                     'DummyOnlineSecurity', 'DummyOnlineBackup',
                     'DummyDeviceProtection', 'DummyTechSupport',
                     'DummyStreamingTV', 'DummyPaperlessBilling',
                     'Bandwidth_GB_Year']]
```

```python
df = churn_df.columns
print(df)
```

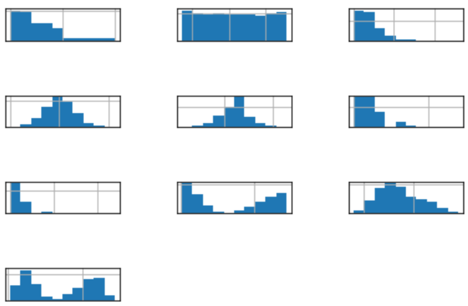
```
Index(['Children', 'Age', 'Income', 'Outage_sec_perweek', 'Email', 'Contacts',
       'Yearly_equip_failure', 'Tenure', 'MonthlyCharge', 'TimelyResponse',
       'Fixes', 'Replacements', 'Reliability', 'Options', 'Respectfulness',
       'Courteous', 'Listening', 'DummyGender', 'DummyChurn', 'DummyTechie',
       'DummyContract', 'DummyPort_modem', 'DummyTablet',
       'DummyInternetService', 'DummyPhone', 'DummyMultiple',
       'DummyOnlineSecurity', 'DummyOnlineBackup', 'DummyDeviceProtection',
       'DummyTechSupport', 'DummyStreamingTV', 'DummyPaperlessBilling',
       'Bandwidth_GB_Year'],
      dtype='object')
```

(DataCamp 2021).

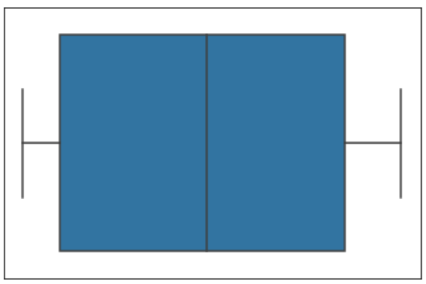
C4. Univariate and Bivariate Visualizations

Univariate Visualizations

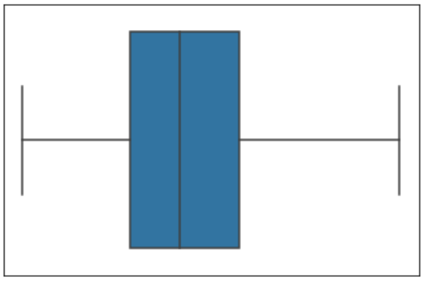
```
# Display histograms of continuous variables
churn_df[['Children', 'Age', 'Income', 'Outage_sec_perweek', 'Email',
          'Contacts', 'Yearly_equip_failure', 'Tenure', 'MonthlyCharge',
          'Bandwidth_GB_Year']].hist()
plt.savefig('churn_pyplot.jpg')
plt.tight_layout()
```



```
# Create corresponding Seaborn boxplots
sns.boxplot('Tenure', data = churn_df)
plt.show()
```

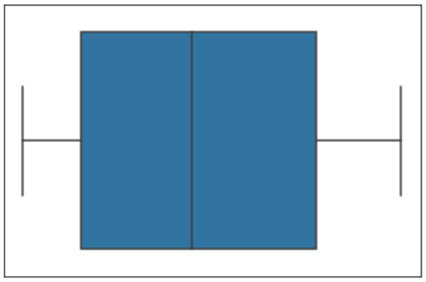


```
sns.boxplot('MonthlyCharge', data = churn_df)
plt.show()
```



(DataCamp 2021).

```
sns.boxplot('Bandwidth_GB_Year', data = churn_df)
plt.show()
```

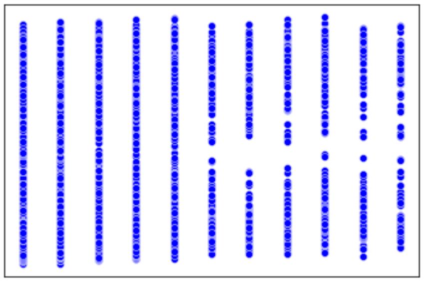


(DataCamp 2021).

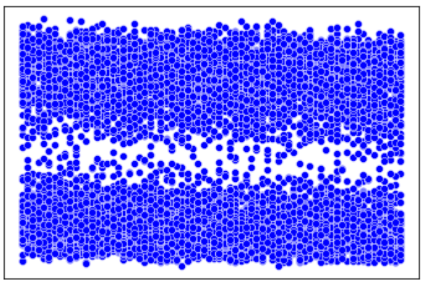
Univariate visualizations confirm there are no outliers or anomalies present in the dataset.

Bivariate Visualizations

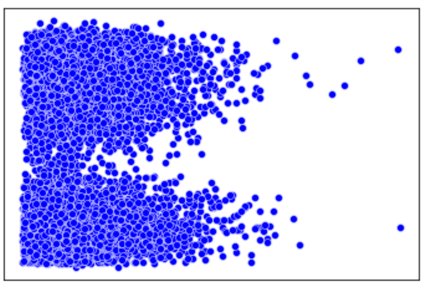
```
# Run scatterplots to show direct or inverse relationships between target & independent variables
sns.scatterplot(x=churn_df['Children'], y=churn_df['Bandwidth_GB_Year'],
color='blue')
plt.show();
```



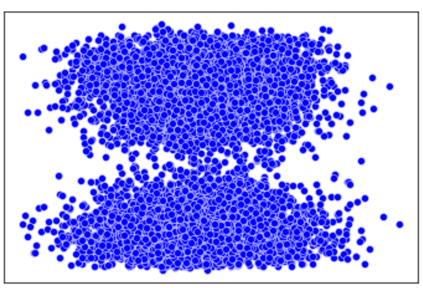
```
sns.scatterplot(x=churn_df['Age'], y=churn_df['Bandwidth_GB_Year'], color='blue')
plt.show();
```



```
sns.scatterplot(x=churn_df['Income'], y=churn_df['Bandwidth_GB_Year'], color='blue')
plt.show();
```

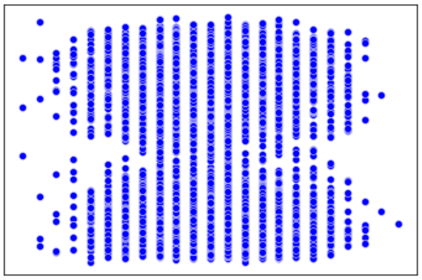


```
sns.scatterplot(x=churn_df['Outage_sec_perweek'], y=churn_df['Bandwidth_GB_Year'], color='blue')
plt.show();
```

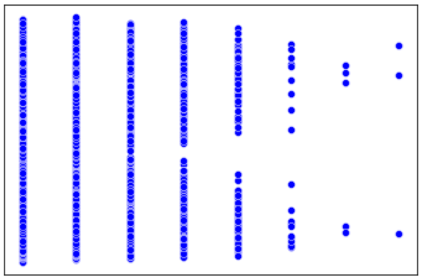


(DataCamp 2021).

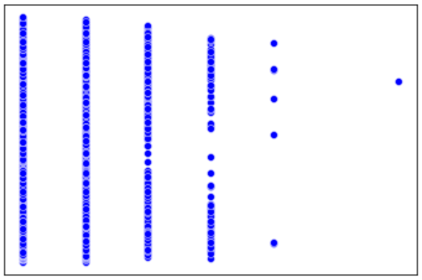
```
sns.scatterplot(x=churn_df['Email'], y=churn_df['Bandwidth_GB_Year'], color='blue')
plt.show();
```



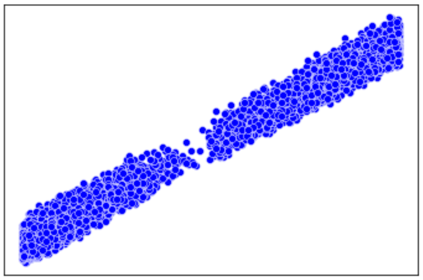
```
sns.scatterplot(x=churn_df['Contacts'], y=churn_df['Bandwidth_GB_Year'], color='blue')
plt.show();
```



```
sns.scatterplot(x=churn_df['Yearly_equip_failure'], y=churn_df['Bandwidth_GB_Year'], color='blue')
plt.show();
```

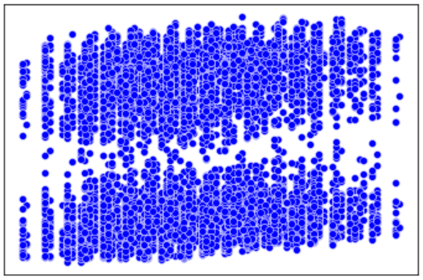


```
sns.scatterplot(x=churn_df['Tenure'], y=churn_df['Bandwidth_GB_Year'], color='blue')
plt.show();
```

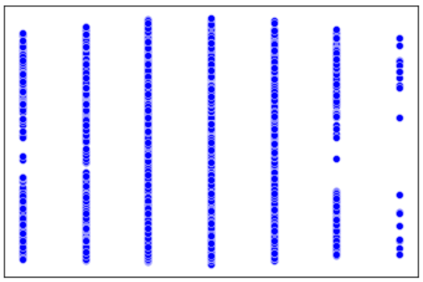


(DataCamp 2021).

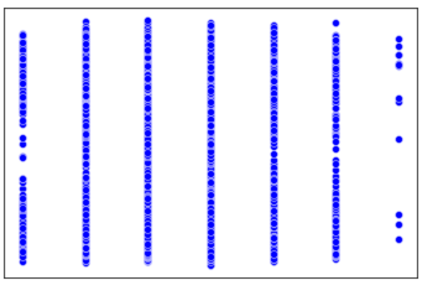
```
sns.scatterplot(x=churn_df['MonthlyCharge'], y=churn_df['Bandwidth_GB_Year'], color='blue')
plt.show();
```



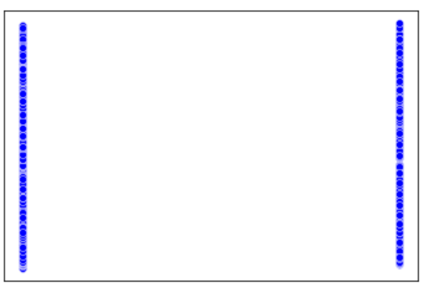
```
sns.scatterplot(x=churn_df['TimelyResponse'], y=churn_df['Bandwidth_GB_Year'], color='blue')
plt.show();
```



```
sns.scatterplot(x=churn_df['Fixes'], y=churn_df['Bandwidth_GB_Year'], color='blue')
plt.show();
```



```
sns.scatterplot(x=churn_df['DummyTechie'], y=churn_df['Bandwidth_GB_Year'], color='blue')
plt.show();
```



(DataCamp 2021).

The bivariate visualizations provide insight into linear relationships of the target variable as well as corresponding predictor variables.

C5. Copy of Cleaned Data Set

```
# Extract cleaned dataset for submission
churn_df.to_csv('churn_prepared.csv')
```

```
churn_df = pd.read_csv('churn_prepared.csv')
df = churn_df.columns
print(df)
```

```
Index(['Unnamed: 0', 'Children', 'Age', 'Income', 'Outage_sec_perweek',
       'Email', 'Contacts', 'Yearly_equip_failure', 'Tenure', 'MonthlyCharge',
       'TimelyResponse', 'Fixes', 'Replacements', 'Reliability', 'Options',
       'Respectfulness', 'Courteous', 'Listening', 'DummyGender', 'DummyChurn',
       'DummyTechie', 'DummyContract', 'DummyPort_modem', 'DummyTablet',
       'DummyInternetService', 'DummyPhone', 'DummyMultiple',
       'DummyOnlineSecurity', 'DummyOnlineBackup', 'DummyDeviceProtection',
       'DummyTechSupport', 'DummyStreamingTV', 'DummyPaperlessBilling',
       'Bandwidth_GB_Year'],
      dtype='object')
```

Part IV: Model Comparison and Analysis

D1. Initial Multiple Regression Model

```
# Initial regression equation to predict Bandwidth_GB_Year,
# using continuous variables
churn_df['intercept'] = 1
lm_bandwidth = sm.OLS(churn_df['Bandwidth_GB_Year'],
                      churn_df[['Children', 'Age', 'Income',
                                'Outage_sec_perweek', 'Email', 'Contacts',
                                'Yearly_equip_failure', 'Tenure',
                                'MonthlyCharge',  'TimelyResponse', 'Fixes',
                                'Replacements', 'Reliability', 'Options',
                                'Respectfulness', 'Courteous', 'Listening',
                                'intercept']]).fit()
print(lm_bandwidth.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:      Bandwidth_GB_Year   R-squared:                       0.989
Model:                            OLS   Adj. R-squared:                  0.989
Method:                 Least Squares   F-statistic:                 5.329e+04
Date:                Fri, 12 Nov 2021   Prob (F-statistic):               0.00
Time:                        14:59:36   Log-Likelihood:                -68489.
No. Observations:               10000   AIC:                         1.370e+05
Df Residuals:                    9982   BIC:                         1.371e+05
Df Model:                          17
Covariance Type:            nonrobust
========================================================================================
                           coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------------
Children                30.9275      1.065     29.050      0.000      28.841      33.014
Age                     -3.3206      0.110    -30.065      0.000      -3.537      -3.104
Income                9.976e-05    8.1e-05      1.231      0.218   -5.91e-05       0.000
Outage_sec_perweek      -0.3501      0.768     -0.456      0.649      -1.856       1.156
Email                   -0.2792      0.755     -0.370      0.712      -1.759       1.201
Contacts                 2.9707      2.312      1.285      0.199      -1.562       7.503
Yearly_equip_failure     0.9080      3.593      0.253      0.801      -6.136       7.952
Tenure                  82.0113      0.086    948.882      0.000      81.842      82.181
MonthlyCharge            3.2768      0.053     61.585      0.000       3.173       3.381
TimelyResponse          -8.8961      3.271     -2.720      0.007     -15.308      -2.484
Fixes                    3.4660      3.064      1.131      0.258      -2.541       9.473
Replacements            -0.1771      2.812     -0.063      0.950      -5.690       5.335
Reliability             -0.2697      2.515     -0.107      0.915      -5.199       4.659
Options                  2.7199      2.611      1.042      0.298      -2.398       7.838
Respectfulness           1.7157      2.689      0.638      0.523      -3.554       6.986
Courteous               -1.3482      2.543     -0.530      0.596      -6.333       3.637
Listening                5.7844      2.420      2.390      0.017       1.040      10.529
intercept               95.8754     26.146      3.667      0.000      44.624     147.127
==============================================================================
Omnibus:                    12280.983   Durbin-Watson:                   1.979
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              968.853
Skew:                           0.449   Prob(JB):                    4.13e-211
Kurtosis:                       1.768   Cond. No.                     5.60e+05
==============================================================================
```

```
churn_df_dummies = churn_df.columns
print(churn_df_dummies)
```

```
Index(['Unnamed: 0', 'Children', 'Age', 'Income', 'Outage_sec_perweek',
       'Email', 'Contacts', 'Yearly_equip_failure', 'Tenure', 'MonthlyCharge',
       'TimelyResponse', 'Fixes', 'Replacements', 'Reliability', 'Options',
       'Respectfulness', 'Courteous', 'Listening', 'DummyGender', 'DummyChurn',
       'DummyTechie', 'DummyContract', 'DummyPort_modem', 'DummyTablet',
       'DummyInternetService', 'DummyPhone', 'DummyMultiple',
       'DummyOnlineSecurity', 'DummyOnlineBackup', 'DummyDeviceProtection',
       'DummyTechSupport', 'DummyStreamingTV', 'DummyPaperlessBilling',
       'Bandwidth_GB_Year', 'intercept'],
      dtype='object')
```

(DataCamp 2021).

```
# Regression Model utilizing all dummy variables
churn_df['intercept'] = 1
lm_bandwidth = sm.OLS(churn_df['Bandwidth_GB_Year'],
                      churn_df[['Children', 'Age', 'Income',
                                'Outage_sec_perweek', 'Email', 'Contacts',
                                'Yearly_equip_failure', 'DummyTechie',
                                'DummyContract', 'DummyPort_modem',
                                'DummyTablet', 'DummyInternetService',
                                'DummyPhone', 'DummyMultiple',
                                'DummyOnlineSecurity', 'DummyOnlineBackup',
                                'DummyDeviceProtection', 'DummyTechSupport',
                                'DummyStreamingTV', 'DummyPaperlessBilling',
                                'Tenure','MonthlyCharge', 'TimelyResponse',
                                'Fixes', 'Replacements','Reliability',
                                'Options','Respectfulness', 'Courteous',
                                'Listening', 'intercept']]).fit()
print(lm_bandwidth.summary())
```

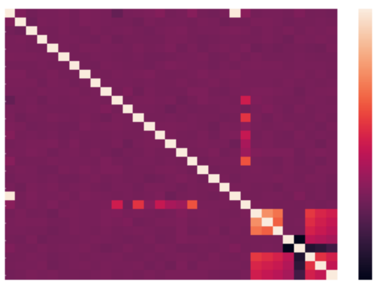
```
                            OLS Regression Results
==============================================================================
Dep. Variable:      Bandwidth_GB_Year   R-squared:                       0.996
Model:                            OLS   Adj. R-squared:                  0.996
Method:                 Least Squares   F-statistic:                 8.675e+04
Date:                Fri, 12 Nov 2021   Prob (F-statistic):               0.00
Time:                        15:03:30   Log-Likelihood:                -63241.
No. Observations:               10000   AIC:                         1.265e+05
Df Residuals:                    9969   BIC:                         1.268e+05
Df Model:                          30
Covariance Type:            nonrobust
=========================================================================================
                            coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------
Children                 30.4177      0.631     48.226      0.000      29.181      31.654
Age                      -3.3153      0.065    -50.671      0.000      -3.444      -3.187
Income                  9.27e-06    4.8e-05      0.193      0.847   -8.48e-05       0.000
Outage_sec_perweek       -0.5259      0.455     -1.156      0.248      -1.418       0.366
Email                     0.1812      0.448      0.405      0.686      -0.696       1.058
Contacts                  2.1263      1.370      1.552      0.121      -0.559       4.811
Yearly_equip_failure      1.2859      2.129      0.604      0.546      -2.887       5.459
DummyTechie               0.6193      3.621      0.171      0.864      -6.478       7.717
DummyContract             3.9328      3.151      1.248      0.212      -2.244      10.110
DummyPort_modem           0.4710      2.707      0.174      0.862      -4.835       5.777
DummyTablet              -1.9813      2.959     -0.670      0.503      -7.781       3.819
DummyInternetService   -373.7111      2.980   -125.411      0.000    -379.552    -367.870
DummyPhone               -2.1515      4.658     -0.462      0.644     -11.282       6.979
DummyMultiple           -76.0773      3.153    -24.130      0.000     -82.257     -69.897
DummyOnlineSecurity      67.4949      2.830     23.850      0.000      61.948      73.042
DummyOnlineBackup       -12.6597      2.931     -4.319      0.000     -18.406      -6.914
DummyDeviceProtection    24.8879      2.807      8.867      0.000      19.386      30.390
DummyTechSupport        -52.5816      2.857    -18.405      0.000     -58.182     -46.981
DummyStreamingTV         30.4799      3.372      9.039      0.000      23.870      37.090
DummyPaperlessBilling    -2.6415      2.752     -0.960      0.337      -8.035       2.752
Tenure                   81.9913      0.051   1600.655      0.000      81.891      82.092
MonthlyCharge             4.7092      0.048     97.416      0.000       4.614       4.804
TimelyResponse           -1.4340      1.939     -0.739      0.460      -5.236       2.368
Fixes                     1.6837      1.817      0.927      0.354      -1.878       5.245
Replacements             -2.4128      1.666     -1.448      0.148      -5.679       0.853
Reliability              -1.5594      1.489     -1.047      0.295      -4.479       1.360
Options                   0.5285      1.547      0.342      0.733      -2.504       3.561
Respectfulness            1.2322      1.593      0.774      0.439      -1.890       4.354
Courteous                 0.4649      1.507      0.308      0.758      -2.490       3.419
Listening                 3.1708      1.434      2.212      0.027       0.361       5.981
intercept                33.1742     16.379      2.025      0.043       1.069      65.280
==============================================================================
Omnibus:                      871.245   Durbin-Watson:                   1.970
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              697.849
Skew:                          -0.559   Prob(JB):                    2.91e-152
Kurtosis:                       2.349   Cond. No.                     5.95e+05
==============================================================================
```

(DataCamp 2021).

D2. Variable Selection Procedure and Evaluation Metric

Justify a statistically based variable selection procedure and a model evaluation metric to reduce the initial model in a way that aligns with the research question.

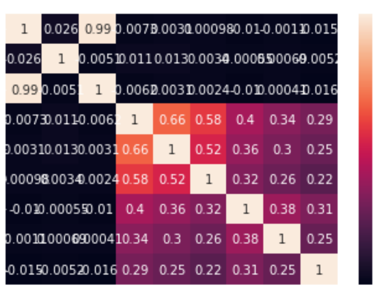
```
 # Dataframe for use in heatmap bivariate analysis of correlation
churn_bivariate = churn_df[['Bandwidth_GB_Year', 'Children', 'Age', 'Income',
                            'Outage_sec_perweek', 'Yearly_equip_failure',
                            'DummyTechie', 'DummyContract', 'DummyPort_modem',
                            'DummyTablet','DummyInternetService', 'DummyPhone',
                            'DummyMultiple','DummyOnlineSecurity',
                            'DummyOnlineBackup', 'DummyDeviceProtection',
                            'DummyTechSupport', 'DummyStreamingTV',
                            'DummyPaperlessBilling','Email', 'Contacts',
                            'Tenure', 'MonthlyCharge', 'TimelyResponse',
                            'Fixes', 'Replacements', 'Reliability',
                            'Options','Respectfulness', 'Courteous',
                            'Listening']]

# Create Seaborn heatmap
sns.heatmap(churn_bivariate.corr(), annot=False)
plt.show()
```



From the initial heatmap, it is hard to gain meaningful insight from the visualization. The purple and darker hues representing variables such as demographics and customer contact options displayed obscure the underlying statistics.

For the following visualization we will once again run the analysis removing these detractors to help see the underlying statistics.

```
 churn_bivariate = churn_df[['Bandwidth_GB_Year', 'Children', 'Tenure',
                             'TimelyResponse', 'Fixes', 'Replacements',
                             'Respectfulness', 'Courteous', 'Listening']]
sns.heatmap(churn_bivariate.corr(), annot=True)
plt.show()
```



With the demographic options removed we can see the numeric representation of the variables remaining.

We can see that retention seems to be a predictor relating to the most variances observed. It is obvious that there is a direct linear relationship between the customer's service period at the telecommunications company and the amount of data consumed.

Since the coefficients of the original OLS model are high (31.0), a multiple linear regression model should be completed on these variables with children greater than or equal to the 0.50 threshold. The p-value for children is 0.000, so it is statistically significant.

The reduced regression equation contains a continuous variable of retention and a child category, and an ordinal independent variables of Timely Fix and Timely Replacement.

For our Regression Analysis we will continue to statistically analyze the following relevant criteria:

"Bandwidth_GB_Year" - The target variable for consideration in the Regression Analysis

"Children" - As noted above, the p-value for children is 0.000, so it is statistically significant.

"Tenure" - Retention of customers is of paramount important to the analysis and must hold its value as a metric.

From the customer survey responses, the following are significant indicators:

"Timely Response" - a strong indicator of customer patience and their perceived importance to the company.

"Timely Fixes" - significant as it represents a period during which customers are without service.

"Timely Replacements" - another significant indicator of customers without service.

"Respectfulness" - perceived value to the company the customer feels and experiences.

"Courteous" - perceived professionalism of the customer service interaction.

"Listening" - Active listening skills demonstrating customer service understanding of issues presented.

D3. Reduced Multiple Regression Model

A Reduced Multiple Regression model including both categorical and continuous variables.

```
# Create a reduced OLS multiple regression
churn_df['intercept'] = 1
lm_bandwidth_reduced = sm.OLS(churn_df['Bandwidth_GB_Year'],
                              churn_df[['Children', 'Tenure', 'Fixes',
                                        'Replacements', 'intercept']]).fit()
print(lm_bandwidth_reduced.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:      Bandwidth_GB_Year   R-squared:                       0.984
Model:                            OLS   Adj. R-squared:                  0.984
Method:                 Least Squares   F-statistic:                 1.537e+05
Date:                Fri, 12 Nov 2021   Prob (F-statistic):               0.00
Time:                        15:12:20   Log-Likelihood:                -70407.
No. Observations:               10000   AIC:                         1.408e+05
Df Residuals:                    9995   BIC:                         1.409e+05
Df Model:                           4
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
Children        31.1763      1.288     24.211      0.000      28.652      33.700
Tenure          81.9518      0.105    783.845      0.000      81.747      82.157
Fixes            1.0728      3.129      0.343      0.732      -5.061       7.206
Replacements    -3.6585      3.149     -1.162      0.245      -9.831       2.514
intercept      506.7695     11.949     42.413      0.000     483.348     530.191
==============================================================================
Omnibus:                      380.733   Durbin-Watson:                   1.978
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              295.369
Skew:                           0.334   Prob(JB):                     7.27e-65
Kurtosis:                       2.488   Cond. No.                         191.
==============================================================================
```

(DataCamp 2021).

Standard error assumes that the covariance matrix of the errors is correctly specified.

By eliminating the non-relevant predictors, our model accounts for 98% of the deviation or variance.

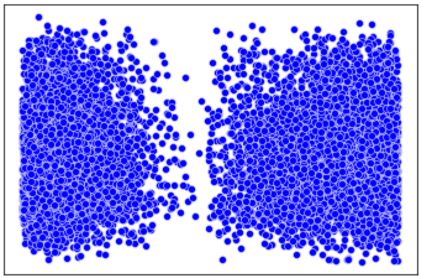
Reduced Multiple Regression Model including four independent variables:

y = 499.67 + 31.14 * Children + 81.85 * Tenure + 1.03 * Fixes - 3.59 * Replacements

E1. Initial and Reduced Model Comparison

```
churn_df = pd.read_csv('churn_prepared.csv')
churn_df['intercept'] = 1
residuals = churn_df['Bandwidth_GB_Year'] - lm_bandwidth_reduced.predict(
    churn_df[['Children', 'Tenure', 'Fixes', 'Replacements','intercept']])
sns.scatterplot(x=churn_df['Tenure'],y=residuals,color='blue')
```

```
<AxesSubplot:xlabel='Tenure'>
```



The Initial Multiple Linear Regression analysis provides information based on 17 continuous variables and 13 categorical variables as follows:

y = 106.74 + 31.59 * Children - 3.27 * Age + 0.00 * Income - 0.27 * Outage_sec_perweek - 0.33 * Email + 2.87 * Contacts + 0.69 * Yearly_equip_failure + 0.65 * DummyTechie + 3.97 * DummyContract + 0.45 * Dummy - Port_modem - 1.99 * DummyTablet - 373.73 * DummyInternetService - 2.18 * DummyPhone - 76.01 * DummyMultiple + 67.72 * DummyOnlineSecurity - 12.93 * DummyOnlineBackup + 24.54 * DummyDeviceProtection - 52.74 * DummyTechSupport + 30.22 * DummyStreamingTV - 2.46 * DummyPaperlessBilling + 82.09 * Tenure + 3.29 * MonthlyCharge - 8.79 * TimelyResponse + 3.38 * Fixes - 0.17 * Replacements - 0.21 * Reliability + 2.69 * Options + 1.74 * Respectfulness - 1.33 * Courteous + 5.74 * Listening

The reduced multiple regression model based on relevant variables we can reach a model expressed as:

y = 499.67 + 31.14 * Children + 81.85 * Tenure + 1.03 * Fixes - 3.59 * Replacements

The reduced regression model consists of the continuous tenure variable and categorical children variable and the ordinal categorical independent variables fixes and replacements.

E2. Output of Performed Calculations

Calculations are included in the above output and diagrams.

E3. Implementation Code

Code is included in each step of the process complete above.

Part V: Data Summary and Implications

F1. Findings and Assumptions Summary

The initial model calculates the R-Squared value as 0.989. Therefore, with only a 1% disparity the model shows evidence of multicollinearity. The relationship can be verified numerically and visually.

Again, retention/tenure seems to be a predictor of most variances. Obviously, there is a linear relationship between a customer's service duration and data volume (GB unit used) at the telecommunications company.

Reducing our linear expression based on relevant variables we can reach a model expressed as:

y = 499.67 + 31.14 * Children + 81.85 * Tenure + 1.03 * Fixes - 3.59 * Replacements

The coefficients suggest that for every 1 unit of:

- Children - Bandwidth_GB_Year will increase 30.14 units
- Fixes - Bandwidth_GB_Year will increase 1.09 units
- Replacements - - Bandwidth_GB_Year will decrease 3.44 units
- Tenure - Bandwidth_GB_Year will increase 80.63 units

The accuracy of the regression model can be increased with access to a larger set and a greater period of historical data.

F2. Recommendation

In order to provide useful insight to shareholders and decision makers this analysis will focus on providing the following information:

With a strong direct linear relationship between the bandwidth used each year (bandwidth_GB_year) and the length of service (tenure) with the telecommunications company, the company should, with all with its power in the field of marketing and customer service, retain satisfied customers. It makes sense to suggest doing this as from the regression model we can observe a relationship showing the longer they stay with the company, the more bandwidth they normally consume. This includes ensuring that customer issues are resolved quickly and that the equipment provided is reliable and of quality, with fewer equipment replacements.

Part VI: Demonstration

G. Panopto Recording

https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=98ea09ee-bee9-481d-9cd4-ade10049c6d3

H. Web Sources

Multiple regression: Python. campus.datacamp.com. (n.d.). Retrieved November 15, 2021, from https://campus.datacamp.com/courses/exploratory-data-analysis-in-python/multivariate-thinking?ex=4.

Essentials of linear regression in python. DataCamp Community. (n.d.). Retrieved November 15, 2021, from https://www.datacamp.com/community/tutorials/essentials-linear-regression-python.

Multiple linear regression with Sklearn and Statsmodels. DataCamp Community. (n.d.). Retrieved November 15, 2021, from https://www.datacamp.com/community/news/multiple-linear-regression-with-sklearn-and-statsmodels-4vner9fxbkm.

Introduction to regression with statsmodels in python course. DataCamp. (n.d.). Retrieved November 15, 2021, from https://www.datacamp.com/courses/introduction-to-regression-with-statsmodels-in-python.

Intermediate regression with statsmodels in python course. DataCamp. (n.d.). Retrieved November 15, 2021, from https://www.datacamp.com/courses/intermediate-regression-with-statsmodels-in-python.

A beginner's guide to linear regression in python with scikit-learn. DataCamp Community. (n.d.). Retrieved November 15, 2021, from https://www.datacamp.com/community/news/a-beginners-guide-to-linear-regression-in-python-with-scikit-learn-nwu7rs1qys.

I. References

Assumptions of multiple linear regression. Statistics Solutions. (2021, August 11). Retrieved November 11, 2021, from https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/assumptions-of-multiple-linear-regression/.

Larose, C. D., & Larose, D. T. (2019). *Data Science using python and R*. Wiley.

Li, L. (2019, February 5). *Introduction to linear regression in python*. Medium. Retrieved November 15, 2021, from https://towardsdatascience.com/introduction-to-linear-regression-in-python-c12a072bedf0.