Adetutu B.

Predictive Modeling - Multiple Linear Regression

Task 1

Part I: Research Question

A1: Research Question

What factors contribute to customers' yearly data usage in GB?

A2: Analysis Goals

This question can help determine and make sense of the aspects that contribute to the average amount of data (in gigabytes) used in a year by our organization's customers. In researching this topic, we can identify trends in customer use habits and craft applicable solutions to help customers maximize their data usage and provide a more seamless data transmission and communication experience when using the company's services.

I plan to use a multiple linear regression model to assess the relationship between the independent variables (multiple factors in the dataset) and my dependent variable (Bandwidth_GB_Year) and make predictions based on the relationship.

Part II: Method Justification

B1: Multiple Linear Regression Model Assumptions

1. ***Linearity:*** For this regression analysis it is important that <u>a linear relationship between the independent and dependent variables exists</u> (Paul, 2018). So, my independent variables (x) should show a linear relationship with my dependent variable (y = Bandwidth_GB_Year). This behavior can be visually inspected using a linear plot where x and y would increase or decrease at a similar rate creating a line of data points.

2. ***Residual Normality:*** The model assumption requires that <u>the residuals of the linear regression should be normally distributed</u>. The residual is the difference between the observed and predicted values of the line of best fit. This assumption checks for a normal distribution of residuals to ensure the validity of our interpretations.

3. ***No Multicollinearity:*** This regression assumes that <u>the independent variables do not have a high correlation to each other</u>. High correlation between independent variables (multicollinearity) could lead to unreliable results and limit the effects of our model, so it needs to be avoided.

4. ***Homoscedasticity:*** This assumption states that <u>there should be a consistent variance in the residuals of all our independent variable values</u>. Meeting this assumption means our dependent variable is properly defined by the independent variables.

B2: Benefits of Python
For this task I will be utilizing Python and importing packages like Pandas, Numpy, Matplotlib, Seaborn, SciPy, statsmodels, and Sklearn. With this language and the associated packages, I can manipulate and transform my data, visualize variate statistics and residuals, and perform statistical calculations/tests to support my linear regression analysis.

B3: Linear Regression Technique Justification
I will be using multiple linear regression to help understand and predict the possible factors that contribute to a customers' yearly data usage. A multiple linear regression requires a continuous target variable, so this technique is the appropriate option as my dependent variable is quantitative and continuous in nature. This technique will provide me with valuable statistical metrics (like R-squared, p-values, etc.) to find an explanation for the variance in data usage and assess the significance of my predictor variables. Not only can I analyze the relationship of different predictors with the target variable, but this technique should help enhance the accuracy of the predictions.

Part III: Data Preparation
C1: Data Cleaning
My goal during the data cleaning process was to ensure that the data is prepared for further analysis. When I initially loaded my data into the environment I used "keep_default_na=False" to keep essential 'None' values within the dataset. I checked for and addressed null values using "isnull()" syntax. I then dropped any columns I would deem unnecessary for/irrelevant to the analysis of my research question using "drop()". I updated column names to follow Python casing rules and clearly represent their description given in the data dictionary using "rename()". Then, I checked for possible outliers in the columns that would be included in my analysis and assessed the maximum and minimum of any suspected column using "nlargest/nsmallest()".

My code for this can be found in section '**C1: Data Cleaning**' in the file named '**D208 Task 1.ipynb**'

C2: Summary Statistics
My dependent (y) variable is the **'bandwidth_gb_year'** column which records the average amount of data customers use per year.

Task 1

Here is an image of its summary statistics using “describe()”:

```
[407]: #Viewing the summary stats for the dependent variable
       df['bandwidth_gb_year'].describe()
```

```
[407]: count    10000.000000
       mean      3392.341550
       std       2185.294852
       min        155.506715
       25%       1236.470827
       50%       3279.536903
       75%       5586.141370
       max       7158.981530
       Name: bandwidth_gb_year, dtype: float64
```

Some of my initial independent (x) variables are numeric while others are categorical. My numeric independent variables consist of the columns: ‘age’, ‘income’, ‘outage_sec_perweek’, ‘tech_support_contacts’, ‘tenure’, ‘monthly_charge’.

Here is an image of their summary statistics I collected using “describe()”:

```
]: #Reviewing the summary stats for numeric independent variables
   print(ind_num.describe())
```

```
                age         income  outage_sec_perweek  tech_support_contacts  \
count  10000.000000   10000.000000        10000.000000           10000.000000
mean      53.078400   39806.926771           10.001848               0.994200
std       20.698882   28199.916702            2.976019               0.988466
min       18.000000     348.670000            0.099747               0.000000
25%       35.000000   19224.717500            8.018214               0.000000
50%       53.000000   33170.605000           10.018560               1.000000
75%       71.000000   53246.170000           11.969485               2.000000
max       89.000000  258900.700000           21.207230               7.000000

             tenure  monthly_charge
count  10000.000000    10000.000000
mean      34.526188      172.624816
std       26.443063       42.943094
min        1.000259       79.978860
25%        7.917694      139.979239
50%       35.430507      167.484700
75%       61.479795      200.734725
max       71.999280      290.160419
```

My categorical independent variables were: ‘gender’, ‘churn’, ‘techie’, ‘contract’, ‘port_modem’, ‘tablet’, ‘internet_service’, ‘phone_service’, ‘multiple_lines’, ‘online_security’, ‘online_backup’, ‘device_protection’, ‘streaming_tv’, ‘streaming_movies’. Since these columns do not have numeric outputs yet, I could not perform statistical mathematical operations to search for the mean, standard deviation, or quartiles here.

Task 1

Instead of summary statistics, here is an image summarizing their categories and proportions:

```
[259]: #Viewing the summary stats for categorical independent variables
vc_list = []
for col in ind_cat.columns:
    counts = ind_cat[col].value_counts().to_dict()
    vc_list.append({col: counts})

vc_list
```

```
[259]: [{'gender': {'Female': 5025, 'Male': 4744, 'Nonbinary': 231}},
 {'churn': {'No': 7350, 'Yes': 2650}},
 {'techie': {'No': 8321, 'Yes': 1679}},
 {'contract': {'Month-to-month': 5456, 'Two Year': 2442, 'One year': 2102}},
 {'port_modem': {'No': 5166, 'Yes': 4834}},
 {'tablet': {'No': 7009, 'Yes': 2991}},
 {'internet_service': {'Fiber Optic': 4408, 'DSL': 3463, 'None': 2129}},
 {'phone_service': {'Yes': 9067, 'No': 933}},
 {'multiple_lines': {'No': 5392, 'Yes': 4608}},
 {'online_security': {'No': 6424, 'Yes': 3576}},
 {'online_backup': {'No': 5494, 'Yes': 4506}},
 {'device_protection': {'No': 5614, 'Yes': 4386}},
 {'streaming_tv': {'No': 5071, 'Yes': 4929}},
 {'streaming_movies': {'No': 5110, 'Yes': 4890}}]
```

That adds up to a total of 1 dependent variable and 20 independent variables prior to data transformation.

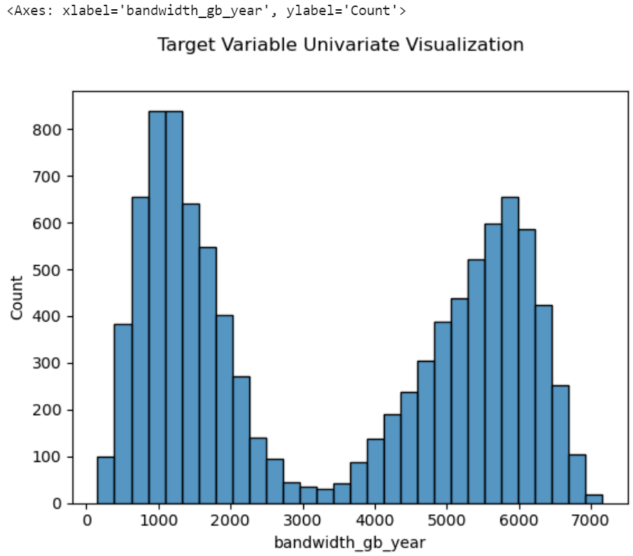
C3: Univariate & Bivariate Visualizations

Along with information about each variable, I generated variate statistics for my dependent variable, independent variables, and the relationship between my dependent and independent variables using various graphing/visualization techniques.

Univariate Statistics for my dependent variable using a histogram:

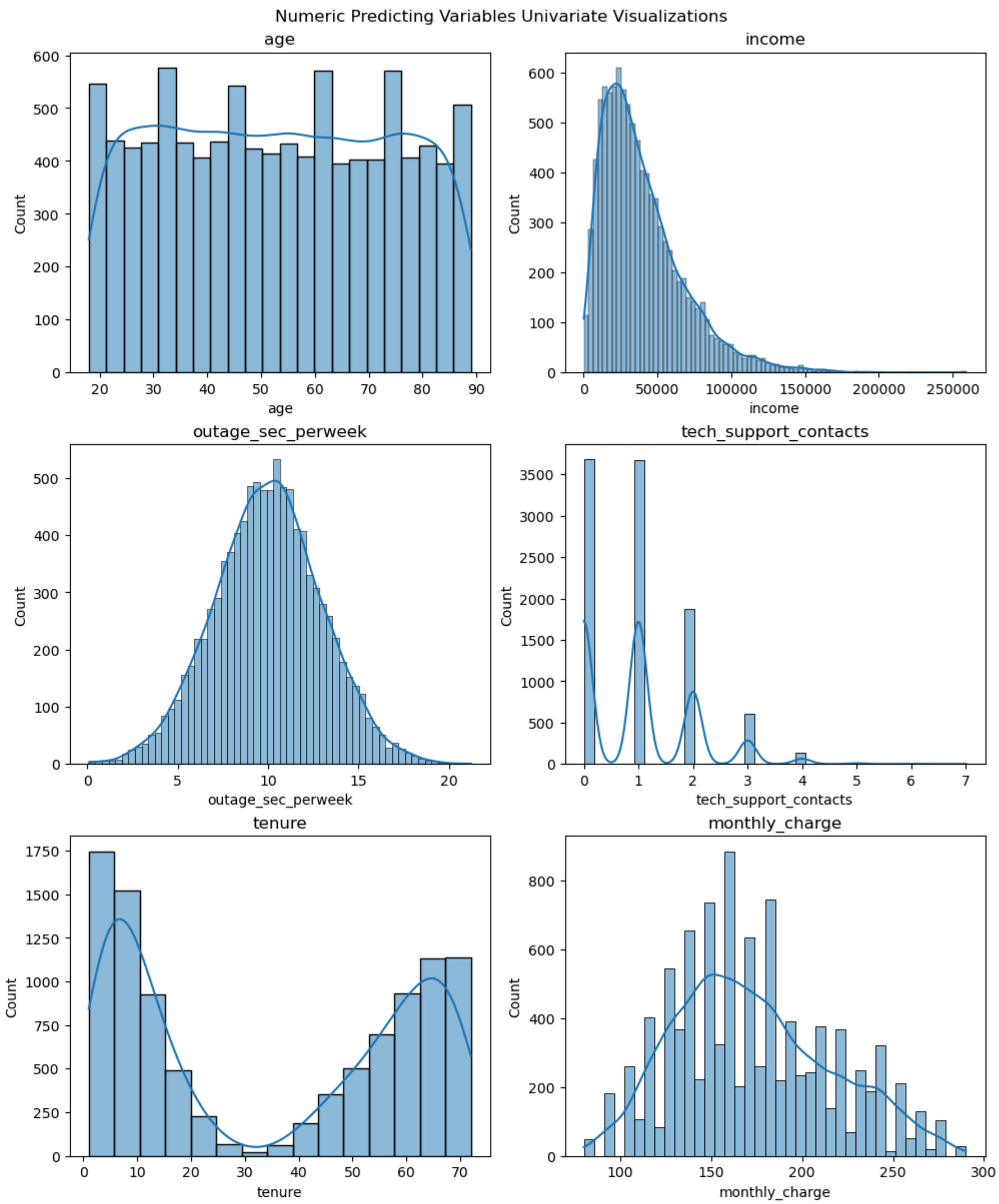
```
[410]: #Creating a histogram visualization for my dependent (target) variable
plt.suptitle('Target Variable Univariate Visualization')
sns.histplot(data=df, x='bandwidth_gb_year', bins=30)
```

```
[410]: <Axes: xlabel='bandwidth_gb_year', ylabel='Count'>
```



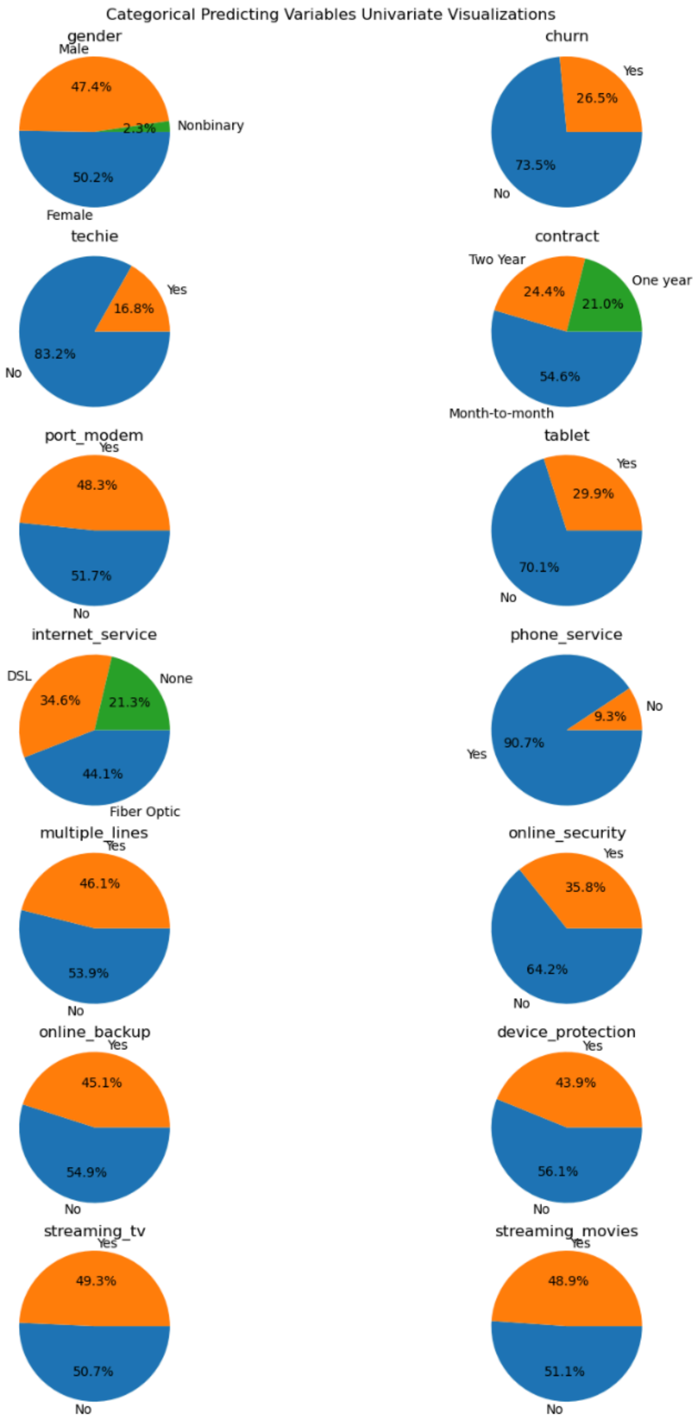
Task 1

Univariate Statistics for my quantitative independent variables using histograms:



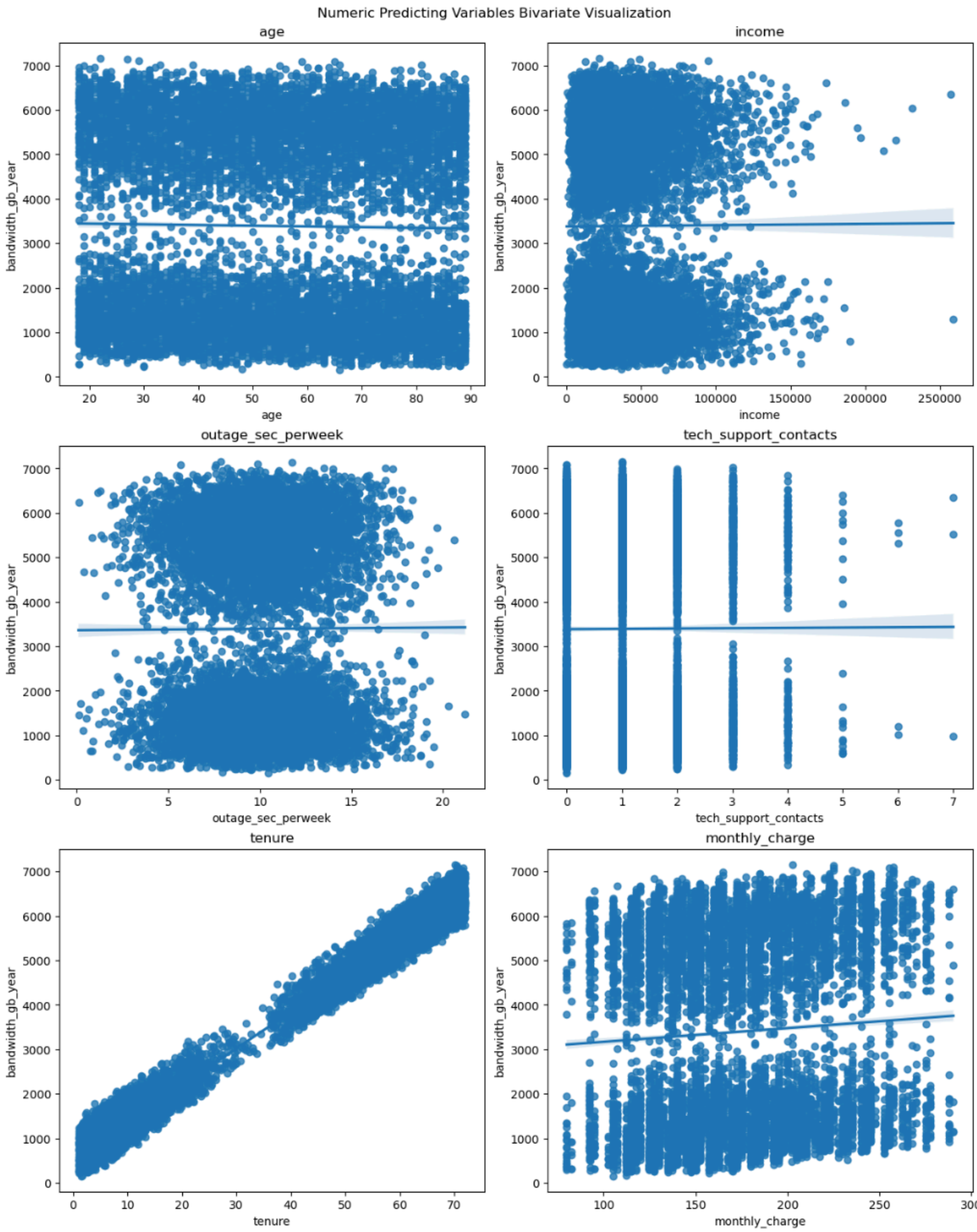
Task 1

Univariate Statistics for my categorical independent variables using pie charts:



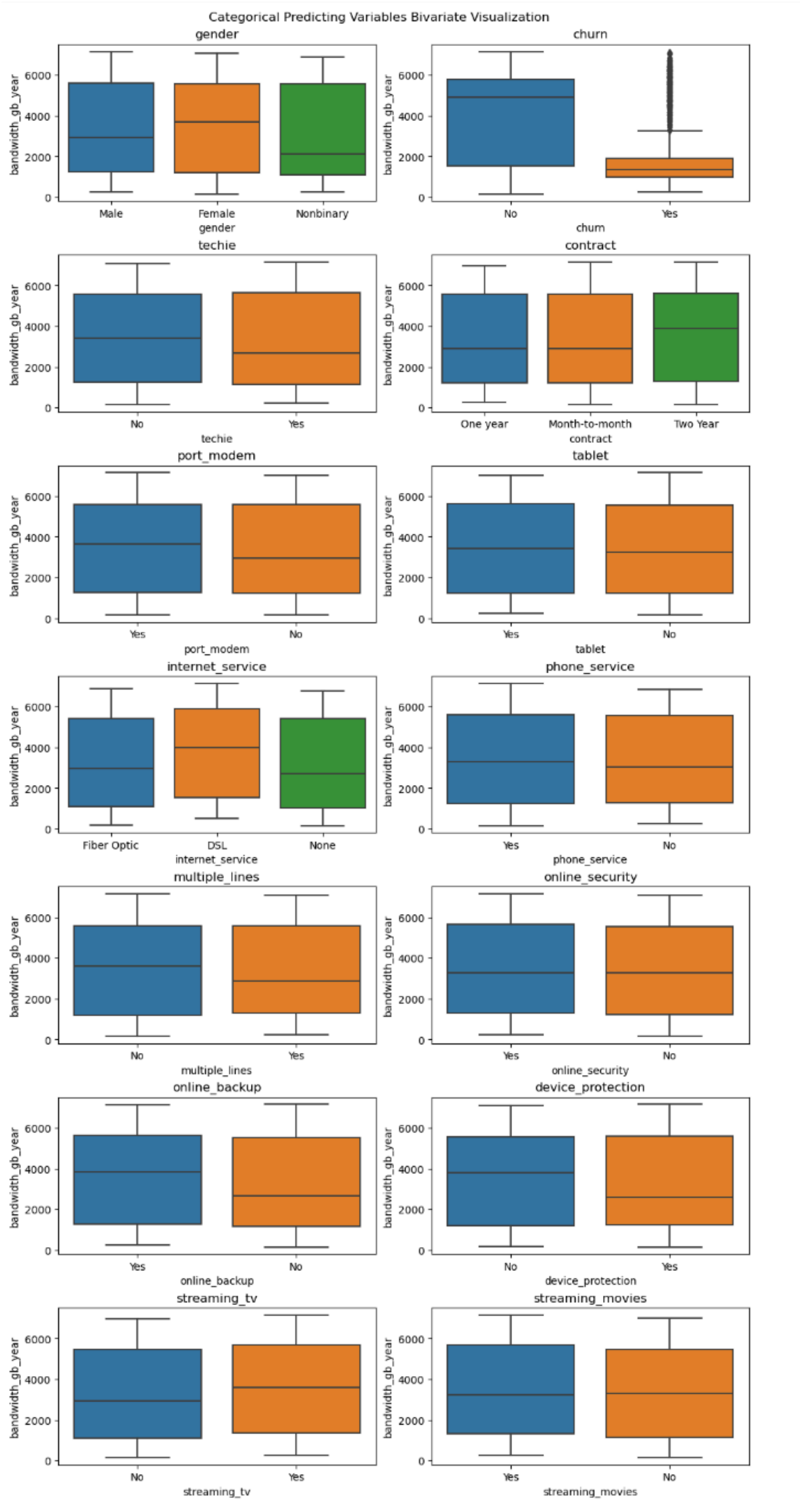
Task 1

Bivariate Statistics visualizing the relationship between my dependent variable and quantitative independent variables with scatterplots:



Task 1

Bivariate Statistics visualizing the relationship between my dependent variable and categorical variables with boxplots:



C4: Data Transformation

In this section, I transformed my categorical variables so their values would read as numeric data. I visually inspected my categorical variables to check for columns with binary (yes/no) columns then replaced them to read as zeros and ones. I then re-expressed all my other nominal data using "get_dummies()" for the categories to produce dummy columns of the variables. In order to follow the k-1 rule for my dummy variables I dropped the first dummy column in gender, contract, and internet_service so I was only left with two dummy categories in each column. So gender_Female, contract_Month-to-month, and internet_service_DSL were all dropped (and will become reference categories), but the other categories (Male/Nonbinary, One/Two years, and Fiber Optic/None) remained. I was now left with 23 total independent variables. I then converted my dummy variables from booleans (True/False) to 'int64' data type using "astype()". This was done to prepare for the statistical modeling/machine learning that would soon take place as numbers–not categories– are required to properly perform this process.

My code for this can be found in section '**C4: Data Transformation'** in the file named '**D208 Task 1.ipynb**'

C5: Data CSV File

My clean data in the csv file named '**clean_churn_data.csv**'

Part IV: Model Comparison and Analysis

D1: Initial Linear Regression Model

I constructed an initial multiple linear regression model by setting my dependent variable and independent variables then creating my regression model using "sm.OLS()" and fitting it with "fit()". My code and resulting output for this initial model can be found in section '**D1: Initial Linear Regression Model'** in the file named '**D208 Task 1.ipynb**'. I will also include my initial model output in section (D3) of this paper for more convenient comparison between the initial and reduced model.

In this initial model I had an R-squared of 0.999 which means that 99.9% of the variation in my dependent variable can be explained by my independent variables. I also had an F-statistic of 4.511e+05 and a BIC of 1.129e+05 which can later be compared to the reduced model to investigate model fit. My condition number was 6.08e+05 which was warned to be quite large. A large condition number may point to signs of multicollinearity which I will investigate in the next section.

D2: Model Reduction Justification

In this section I chose to reduce my independent variables and only select features that showed statistical significance (a p-value < 0.05). Rather than looking at 23 variables with varying relationships with the dependent variable, I wanted to select features with an observed relationship to the bandwidth_gb_year. I performed backward eliminations and individually removed features with the least significance in the model. This selection method left me with the 12 selected features: 'age', 'tenure', 'monthly_charge', 'gender_Male', 'gender_Nonbinary', 'internet_service_Fiber Optic', 'internet_service_None', 'online_security', 'online_backup', 'device_protection', 'streaming_tv', 'streaming_movies' (the constant is not an independent variable). However, this feature selection method does not address possible multicollinearity issues, so I followed up by assessing the Variance Inflation Factor (VIF) of my selected features. My output showed that all of my selected features had a VIF below 5 (though monthly_charge was at about VIF ≈ 4.9). VIFs above 5 suggest mild to extreme multicollinearity issues, but I chose to keep monthly_charge which was close to a VIF of 5 because of its statistical significance. While monthly_charge could show moderate multicollinearity, removing the feature from my model worsened the fit. In order to ensure that my model reduction did not lead to a weaker fit, I chose to keep all 12 features as the multicollinearity was not too extreme and monthly_charge seemed statistically impactful to the analysis.

D3: Reduced Linear Regression Model

I have included the output of my initial model and the reduced model following feature selection. I will compare the two in the next section.

Image of the initial model output:

[288]:

OLS Regression Results

Dep. Variable:	bandwidth_gb_year	**R-squared:**	0.999
Model:	OLS	**Adj. R-squared:**	0.999
Method:	Least Squares	**F-statistic:**	4.511e+05
Date:	Wed, 30 Oct 2024	**Prob (F-statistic):**	0.00
Time:	17:38:45	**Log-Likelihood:**	-56344.
No. Observations:	10000	**AIC:**	1.127e+05
Df Residuals:	9976	**BIC:**	1.129e+05
Df Model:	23		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
age	-3.3759	0.033	-102.937	0.000	-3.440	-3.312
income	2.217e-05	2.41e-05	0.921	0.357	-2.5e-05	6.93e-05
churn	2.8872	2.147	1.345	0.179	-1.321	7.096
outage_sec_perweek	0.0493	0.228	0.216	0.829	-0.398	0.496
tech_support_contacts	-1.2331	0.687	-1.796	0.073	-2.579	0.113
techie	-1.3065	1.822	-0.717	0.473	-4.878	2.265
port_modem	0.9626	1.358	0.709	0.478	-1.699	3.624
tablet	0.1329	1.483	0.090	0.929	-2.774	3.040
phone_service	-0.6214	2.335	-0.266	0.790	-5.198	3.955
multiple_lines	1.1608	2.471	0.470	0.639	-3.683	6.005
online_security	70.9132	1.428	49.648	0.000	68.113	73.713
online_backup	47.2486	1.978	23.889	0.000	43.372	51.126
device_protection	59.7532	1.582	37.779	0.000	56.653	62.854
streaming_tv	138.8625	3.004	46.226	0.000	132.974	144.751
streaming_movies	101.4804	3.583	28.320	0.000	94.456	108.505
tenure	81.9448	0.031	2648.879	0.000	81.884	82.005
monthly_charge	2.0773	0.064	32.463	0.000	1.952	2.203
gender_Male	65.6324	1.375	47.735	0.000	62.937	68.328
gender_Nonbinary	-21.3311	4.567	-4.670	0.000	-30.284	-12.378
contract_One year	3.8246	1.814	2.108	0.035	0.268	7.381
contract_Two Year	4.3457	1.730	2.513	0.012	0.955	7.736
internet_service_Fiber Optic	-454.4823	2.006	-226.614	0.000	-458.414	-450.551
internet_service_None	-385.7334	2.046	-188.535	0.000	-389.744	-381.723
const	441.5382	7.221	61.150	0.000	427.385	455.692

Omnibus:	2194.575	**Durbin-Watson:**	1.991
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	4381.934
Skew:	1.318	**Prob(JB):**	0.00
Kurtosis:	4.888	**Cond. No.**	6.08e+05

Task 1

Image of the reduced model output:

[301]:

OLS Regression Results

Dep. Variable:	bandwidth_gb_year	**R-squared:**	0.999
Model:	OLS	**Adj. R-squared:**	0.999
Method:	Least Squares	**F-statistic:**	8.644e+05
Date:	Wed, 30 Oct 2024	**Prob (F-statistic):**	0.00
Time:	17:45:16	**Log-Likelihood:**	-56351.
No. Observations:	10000	**AIC:**	1.127e+05
Df Residuals:	9987	**BIC:**	1.128e+05
Df Model:	12		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
age	-3.3775	0.033	-103.025	0.000	-3.442	-3.313
online_security	70.8045	1.419	49.900	0.000	68.023	73.586
online_backup	46.4866	1.574	29.535	0.000	43.401	49.572
device_protection	59.3852	1.437	41.323	0.000	56.568	62.202
streaming_tv	137.9252	2.006	68.750	0.000	133.993	141.858
streaming_movies	100.1848	2.278	43.983	0.000	95.720	104.650
tenure	81.9227	0.026	3190.831	0.000	81.872	81.973
monthly_charge	2.1155	0.035	60.489	0.000	2.047	2.184
gender_Male	65.6917	1.374	47.822	0.000	62.999	68.384
gender_Nonbinary	-21.3025	4.567	-4.664	0.000	-30.255	-12.350
internet_service_Fiber Optic	-455.5149	1.683	-270.669	0.000	-458.814	-452.216
internet_service_None	-385.4753	1.931	-199.624	0.000	-389.260	-381.690
const	440.8402	4.544	97.020	0.000	431.933	449.747

Omnibus:	2197.368	**Durbin-Watson:**	1.991
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	4389.575
Skew:	1.320	**Prob(JB):**	0.00
Kurtosis:	4.889	**Cond. No.**	1.40e+03

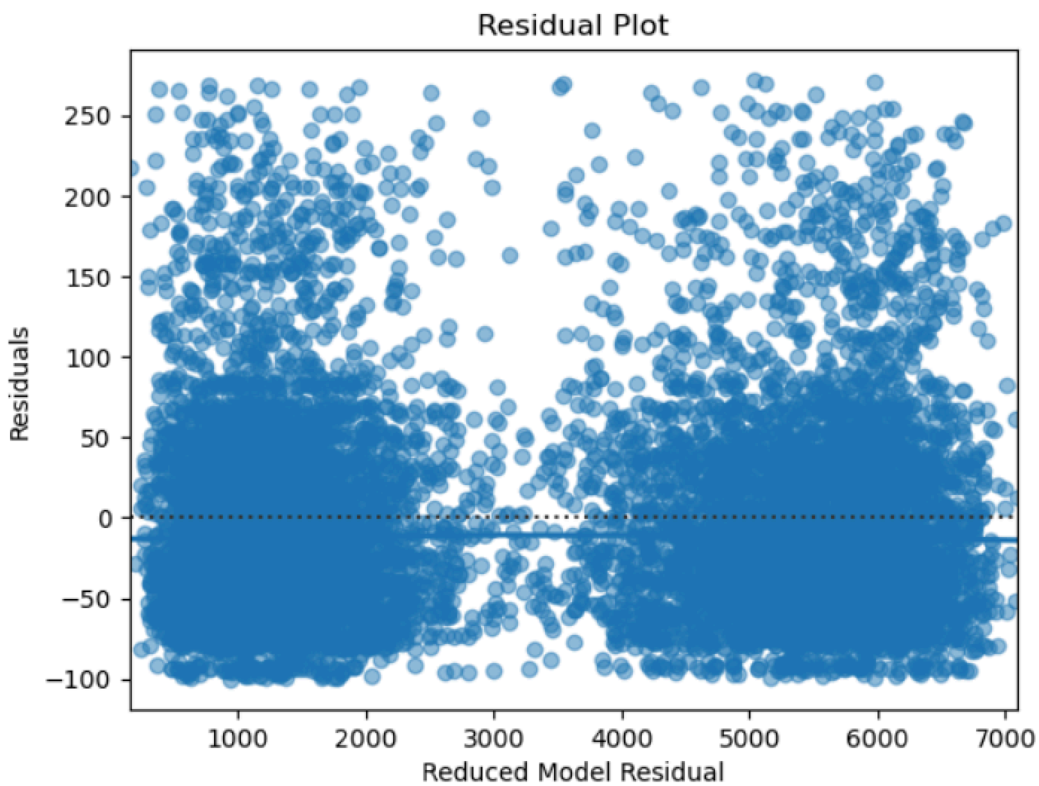
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.4e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

E1: Comparing Initial & Reduced Model

Comparing the initial and reduced model there is a noticeable reduction in the total features used in the regression. As previously stated, I went from 23 independent variables to 12 independent variables. The R-squared and Adjusted R-squared remained the same 0.999 (99.9%) between the two models. So did the AIC which remained at 1.127e+05 in both models. However, the **evaluation metric BIC** decreased from 1.129e+05 in the initial model to 1.128e+05 in the reduced model. Though the change was very small this metric helps compare the goodness of fit for models and the model with the smallest fit (reduced model) is assumed to be a better fit for the data. There was also a noticeable decrease in the condition number in the reduced model going down to 1.40e+03 from its initial value of 6.61e+05. While the number is still a bit on the large side, I have already explained my decision to keep the "monthly_charge" feature as removing it would decrease the condition number but worsen the fit of the model.

E2: Analysis Output & Calculations

Here is the output residual plot and residual standard error of my reduced model.




```
[285]: #Checking the residual standard error of the reduced model
residuals = regres_model_results.resid
mse = np.mean(residuals**2)
reduced_rse = np.sqrt(mse)

print("Residual Std Err")
print('reduced model:', reduced_rse)
```

```
Residual Std Err
reduced model: 67.77048826737139
```

Task 1

E3: Executable Code (error-free)
All my code can be found in the same in the same file titled: '**D208 Task 1.ipynb**'

Part V: Data Summary and Implications
F1: Analysis Results
The equation from multiple linear regression analysis is as follows:
y = 440.8402 - 3.3775(**age**) + 81.9227(**tenure**) - 21.3025(**gender_Nonbinary**) + 65.6917(**gender_Male**) - 455.5149(**internet_service_Fiber Optic**) - 385.4753(**internet_service_None**) + 2.1155(**monthly_charge**) + 70.8045(**online_security**) + 46.4866(**online_backup**) + 59.3852(**device_protection**) + 137.9252(**streaming_tv**) + 100.1848(**streaming_movies**)

Using the coefficients in this equation I can describe the behavior of each predictor variable. Keeping all things constant:

- An additional year increase in age predicts a decrease of 3.3775 gigabytes in a customer's yearly data usage.
- An additional month increase in a customer's tenure predicts an increase of 81.9227 gigabytes in a customer's yearly data usage.
- If an individual is identified as nonbinary (compared to the reference category, which is female) there is a predicted decrease of 21.3025 gigabytes in a customer's yearly data usage.
- However, if the individual is identified as male (compared to the female reference category) there is a predicted increase of 65.6917 gigabytes in the customer's yearly data usage.
- If a customer has fiber optic internet service (compared to DSL) there is a predicted decrease of 455.5149 gigabytes of data usage in a year.
- If the customer does not have internet service (compared to DSL) there is a predicted decrease of 385.4753 gigabytes of data used yearly.
- A one unit increase in a customer's monthly charge predicts increased yearly data usage of 2.1155 gigabytes.
- If the customer has online security (compared to not having it) there is a predicted increase of 70.8045 gigabytes used per year.
- If the customer has online backup (compared to not having it) there is a predicted increase of 46.4866 gigabytes used per year.
- If the customer has device protection (compared to not having it) there is a predicted increase of 59.3852 gigabytes used per year.
- If the customer has streaming tv (compared to not having it) there is a predicted increase of 137.9252 gigabytes used per year.

- If the customer has streaming movies (compared to not having it) there is a predicted increase of 100.1848 gigabytes used per year.

In order to assess the statistical significance of my regression, I looked at the probability of my f-statistic which was zero (prob(F-statistic) = 0.00). This value being below 0.05 implies that my regression does have statistical significance, and my independent variables are statistically meaningful. In terms of practical significance, I believe that most of the independent variables tied to this regression are difficult to directly alter by the company themselves. However, the company can use different combinations of this information to target specific markets/demographics and services to place more emphasis and promotion power on. I will elaborate on this further in the next section (F2).

Limitations:
I believe one limitation to this analysis lies in the timeframe inconsistency among the variables. While my dependent variable aggregates data over a year, some independent variables like tenure and monthly charge are measured using different time frames. This could have led to varying interpretations if my regression was used to test different time frames. For example, weekly or monthly changes may show immediate changes (surges or declines) in data usage, but the impact on yearly data usage could manifest differently over a longer time period.

Another limitation in this analysis was the presence of multicollinearity in my independent variable, while all of my VIF values were below 5 (which generally suggest moderate multicollinearity) I still had a warning that my condition number was large after reduction. Initially, I considered removing the monthly_charge variable from the model as it was very close to a VIF of 5. However, after conducting a model fit assessment, I found that excluding this variable worsened the performance of the model (seen through multiple evaluating metrics). I concluded that 'monthly_charge' was valuable predictive information for my analysis, but the presence of multicollinearity may impact the stability and interpretability of the coefficient estimates (especially the 'monthly_charge' estimate). Even with these concerns I still believe the analysis remains informative for understanding the factors influencing the yearly data usage of the customers.

F2: Next Course of Action
Looking at the coefficient estimates of this analysis, it seems that independent variables like **internet_service_Fiber Optic** (-455.5149), and **internet_service_None** (-385.4753), **streaming_tv** (137.9252), **streaming_movies** (100.1848) have the largest impact on yearly customer data usage. Based on this information, the company can actively promote these streaming services and add-on features by creating marketing campaigns to showcase the benefits of their entertainment and internet services (especially DSL) and encourage an uptick in

customer data usage. They can also offer discounts or promotions for bundled internet access and streaming services, as the streaming features could mitigate the decrease in data usage for customers who use/prefer Fiber Optic or no internet service over DSL. While variables like **tenure** and **online security** had lower coefficient estimates than the aforementioned features, they still had a statistically significant impact on **'bandwidth_gb_year'**. Regularly analyzing patterns in customer data usage and customer feedback and offering data support plans catered to heavy data users or adjusting what services you offer based on data consumption trends can aid in customer retention and loyalty by providing for customers based on their needs.

I would also suggest exploring methods to mitigate multicollinearity and standardize the timeframes for the variables in future research. Exploring external factors (such as sales tactics) or seasonal trends (like holidays or major sports seasons) that could influence the independent variables and data usage patterns may also enhance the robustness of the model.

Part VI: Demonstration

G: Code Web Sources

Middleton, K. (n.d.). Dr. Middleton PA Step-by-Step Guide (NBM3).

H: In-Text Citations

Paul, S. (2018, October 31). *Essentials of linear regression in python*. DataCamp. https://www.datacamp.com/tutorial/essentials-linear-regression-python