Task 2

Adetutu B.

Predictive Modeling - Logistic Regression

Part I: Research Question

A1: Research Question

What factors contribute to customer churn?

A2: Analysis Goals

I will use this question to investigate the factors that contribute to customer churn. Analyzing this research topic can help us determine the variables that contribute to a customer's decision to discontinue their services with the company. We can then use our insights to implement relevant strategies that improve customer retention and minimize the costs of needing to acquire a larger pool of new customers.

This time, I will be using a logistic regression model to make predictions on my categorical dependent variable (Churn) by examining its relationship to independent variables in my data set.

Part II: Method Justification

B1: Logistic Regression Model Assumptions

1. ***Binary:*** This assumption requires that the dependent variable is binary. My dependent variable, Churn, fits this assumption as it is represented as "yes/no" values.

2. ***Independent Observations:*** This regression requires that no two observations should be repeated. Each observation should be independent of one another.

3. ***Linearity:*** Similar to multiple linear regression, logistic regression requires no multicollinearity and assumes linearity of independent variables. However, for this regression type we also assume the log odds of the dependent variable. So while a linear relationship between the independent and dependent variables is not required, the independent variables should be linearly related to the log odds of the dependent variable.

4. ***Sample Size:*** This regression also requires a large sample size. The sample size needs to be big enough to ensure that the conclusions we draw have validity (Li, 2019).

B2: Benefits of Python

I will be using Python to complete this task and support my regression analysis. The packages I will use include Pandas, Numpy, Seaborn, Matplotlib, SciPy, statsmodels, and sklearn. I chose Python as it should be able to handle my fairly large dataset without issue. With the libraries and their offered features I will be able to wrangle my data, create arrays, build machine learning models, run statistical tests/analyses, and visualize my data.

B3: Logistic Regression Technique Justification

I will be analyzing my research question using logistic regression modeling because my dependent variable (churn) is categorical binary variable. Utilizing this technique will enable me to describe and explore the relationship between my categorical dependent variable and its possible independent variables. Using this method will also allow me to estimate the effects of these independent variables and make predictions and future suggestions related to these variables to improve customer churn rates.

Part III: Data Preparation

C1: Data Cleaning

For the data cleaning portion, I updated the names for my columns using "rename()". I wanted the column names to fit proper casing rules and be intuitively titled so they matched well with their data dictionary descriptions. After updating the python-casing, I dropped any columns that would not be important to my analysis. I then checked for null values and possible outliers using code "isnull()" and "describe()". I had no null values and found nothing of major concern regarding outliers. Finally, before moving on to my summary statistics, I wanted to split up my independent variables into a quantitative and qualitative group so I would have a simpler time handling the data. I grouped categorical independent variables under 'ind_cat' and numeric independent variables under 'ind_num' by using "select_dtypes()" to exclude any data types that did not belong in each group.

My annotated code can be found under section '**C1: Data Cleaning**' in the file titled '**D208 Task 2.ipynb**'.

C2: Summary Statistics

My dependent (y) variable is the **'churn'** column. This column records whether or not a customer discontinued their service within the last month. As this is a logistic regression, my dependent variable is categorical. Since my data has not been transformed yet, I cannot provide numeric values like the min, max, mean of this column. However, I can provide the amount of values that fall into each category within the column.

Dependent variable summary description using "value_counts()":

```
[519]: # Dependent variable value summary
       df['churn'].value_counts()
```

```
[519]: churn
       No     7350
       Yes    2650
       Name: count, dtype: int64
```

Similar to my dependent variable, some of my independent variables are also categorical. This includes the columns ‘area_type’, ‘gender’, ‘techie’, ‘contract’, ‘port_modem’, ‘internet_service’, ‘phone_service’, ‘multiple_lines’, ‘online_security’, ‘online_backup’, ‘device_protection’, ‘tech_support’, ‘streaming_tv’, and ‘streaming_movies’. (14)

I also used “value_counts()” for these categorical independent variables:

```
vc_list
```

```
[{'area_type': {'Suburban': 3346, 'Urban': 3327, 'Rural': 3327}},
 {'gender': {'Female': 5025, 'Male': 4744, 'Nonbinary': 231}},
 {'techie': {'No': 8321, 'Yes': 1679}},
 {'contract': {'Month-to-month': 5456, 'Two Year': 2442, 'One year': 2102}},
 {'port_modem': {'No': 5166, 'Yes': 4834}},
 {'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}},
 {'tech_support': {'No': 6250, 'Yes': 3750}},
 {'streaming_tv': {'No': 5071, 'Yes': 4929}},
 {'streaming_movies': {'No': 5110, 'Yes': 4890}}]
```

As for my remaining independent variables: ‘age’, ‘income’, ‘outage_sec_perweek’, ‘tech_support_contacts’, ‘yearly_equip_failure’, ‘tenure’, ‘monthly_charge’, and ‘bandwidth_gb_year’ I was able to use “describe()” as these were quantitative data types. (8)

Numeric independent variables:

```
# Summary stats for quantitative 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

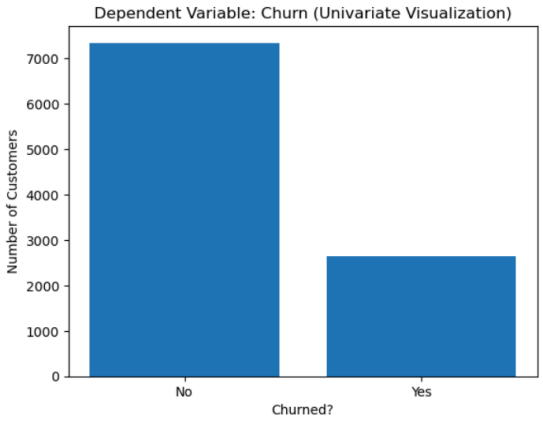
	yearly_equip_failure	tenure	monthly_charge	bandwidth_gb_year
count	10000.000000	10000.000000	10000.000000	10000.000000
mean	0.398000	34.526188	172.624816	3392.341550
std	0.635953	26.443063	42.943094	2185.294852
min	0.000000	1.000259	79.978860	155.506715
25%	0.000000	7.917694	139.979239	1236.470827
50%	0.000000	35.430507	167.484700	3279.536903
75%	1.000000	61.479795	200.734725	5586.141370
max	6.000000	71.999280	290.160419	7158.981530

That is a total of 1 dependent variable and 22 independent variables prior to data transformation.

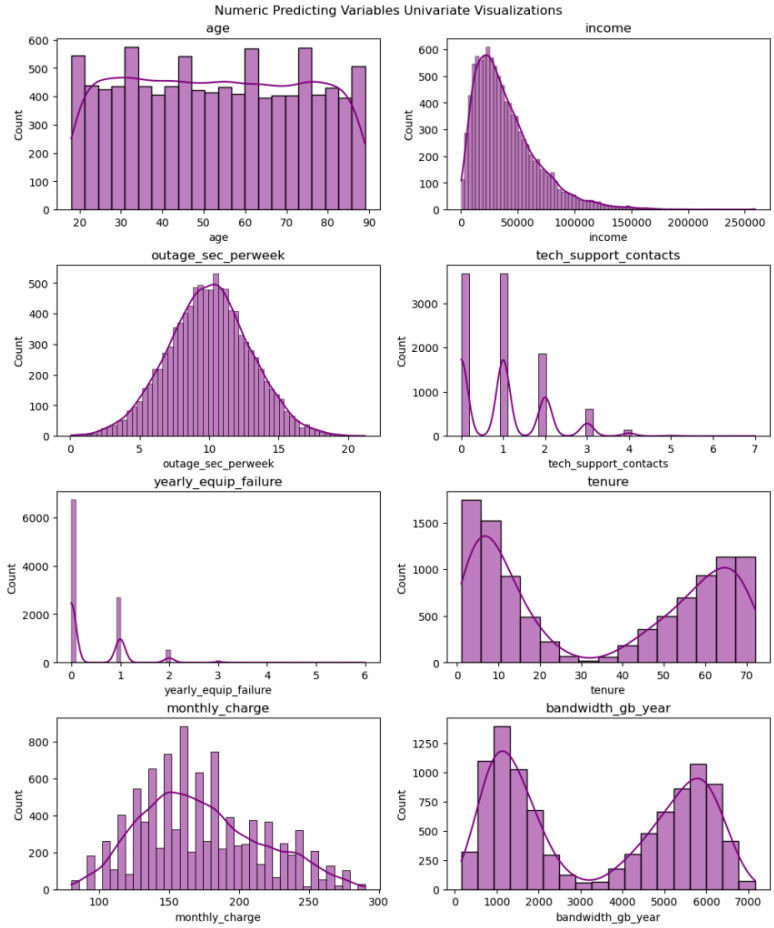
C3: Univariate & Bivariate Visualizations

Here are my variate statistics visualizations for my dependent variable, independent variables, and the relationship my independent variables have with churn

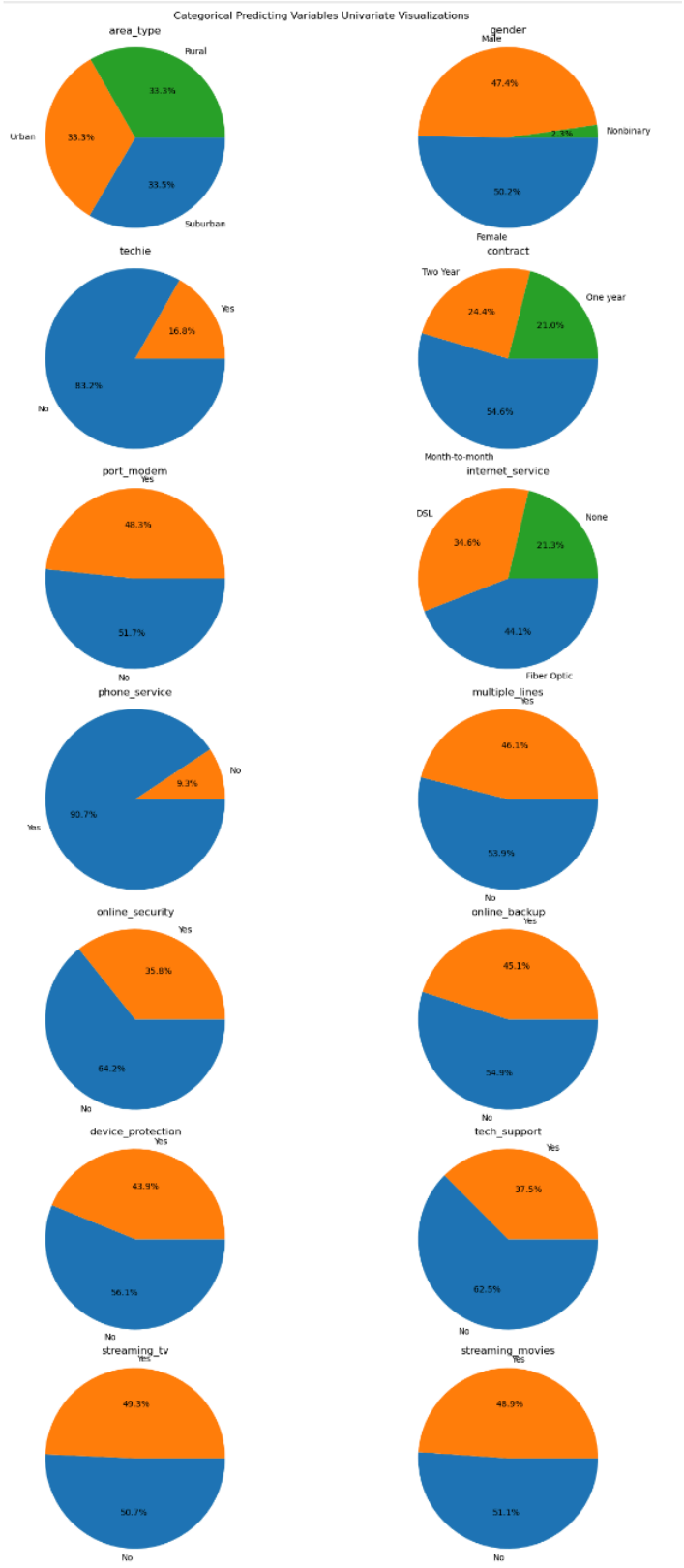
Univariate Statistics for the dependent variable using a bar graph:



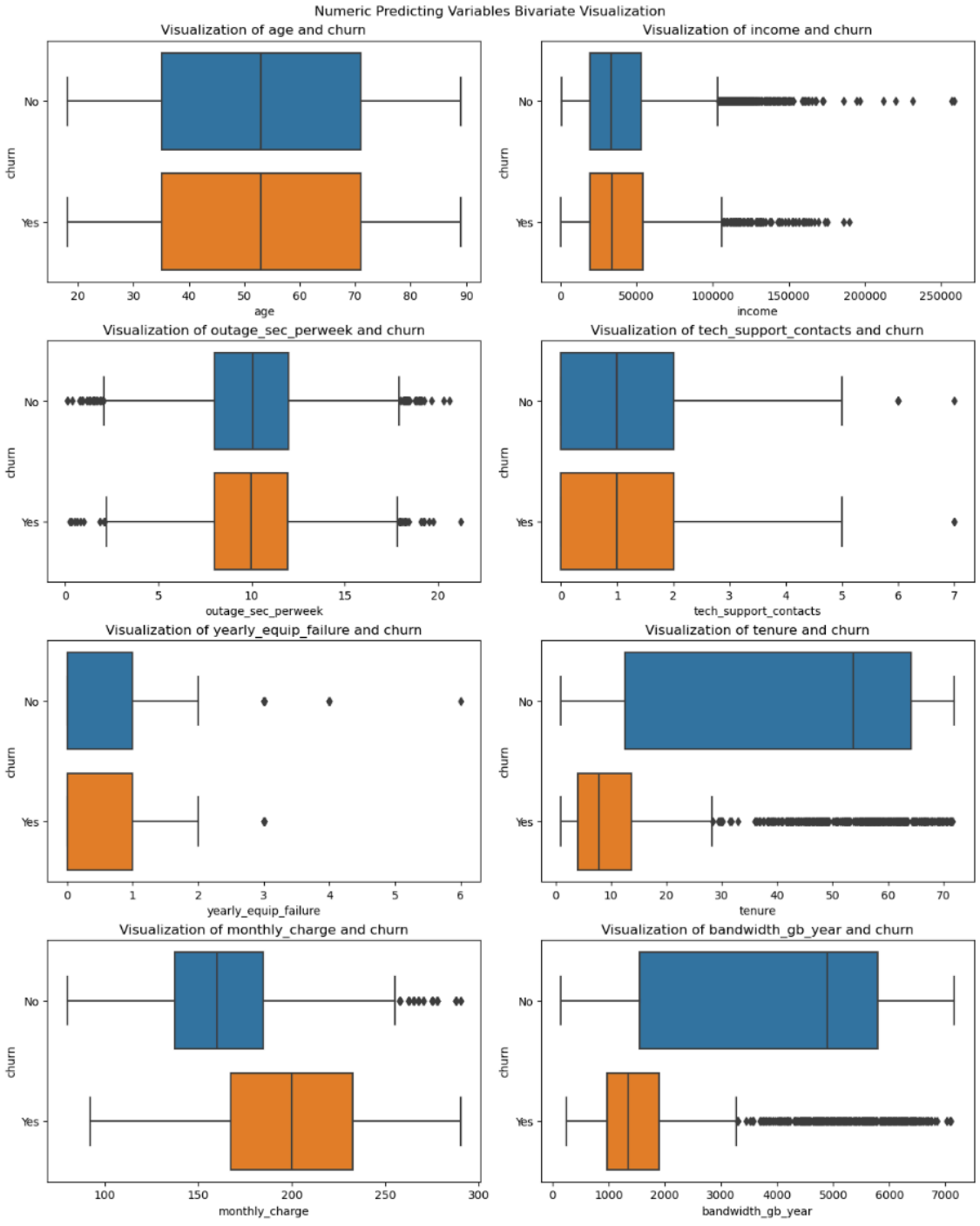
Univariate Statistics for my quantitative independent variables using histograms:



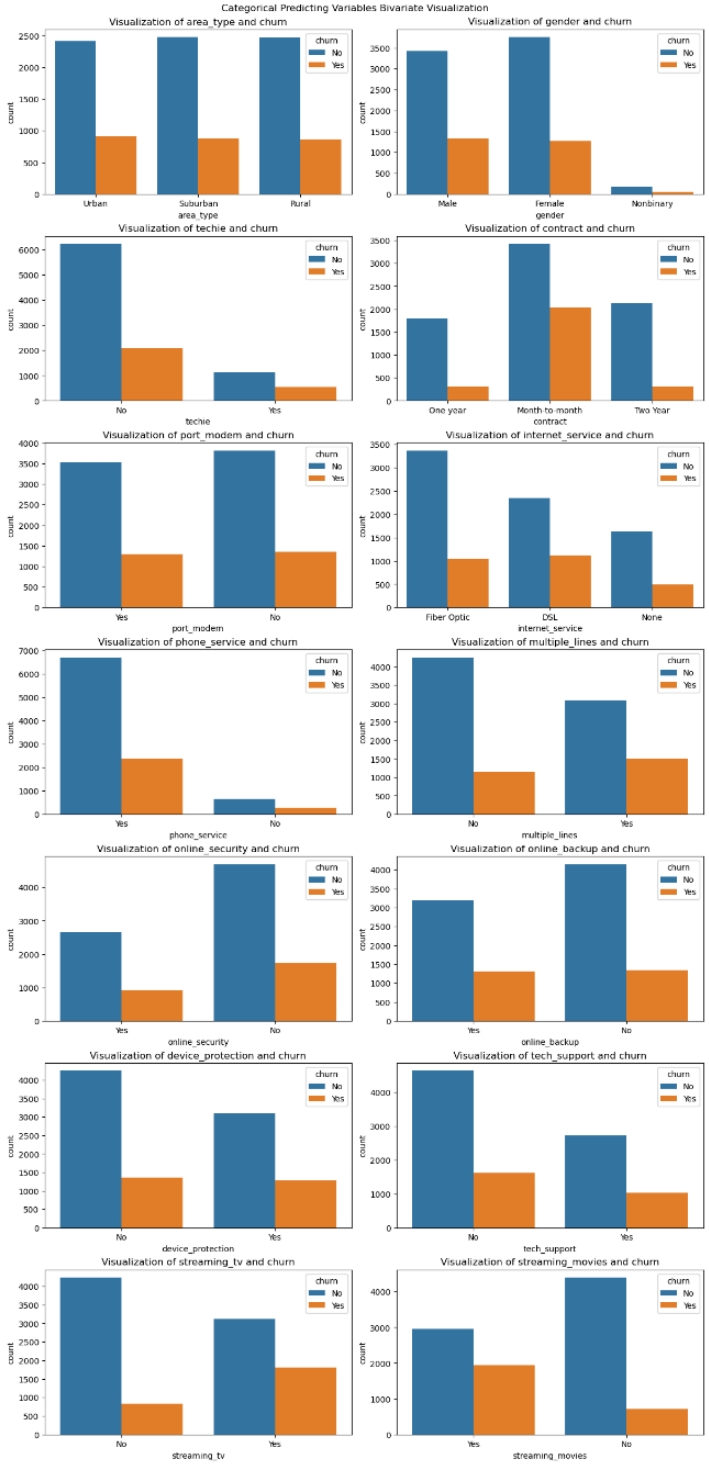
Univariate Statistics for my categorical independent variables using pie charts:



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



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



C4: Data Transformation

To transform my data I began by inspecting my independent variables. As this was a regression, I needed all of my categorical columns to also read as quantitative variables. I encoded all of my binary (yes/no) variables–both independent and dependent–to be expressed where every No = 0 and Yes = 1. I followed by one hot encoding the rest of my independent variables ('area_type, gender, contract', and 'internet_service') by using "get_dummies()" to create dummy variables/columns to be used in the analysis. I also dropped one column in the variables I one hot encoded using "drop_first()" to mitigate multicollinearity. Afterward, I converted my dummy variables (area_type_Suburban, area_type_Urban, gender_Male, gender_Nonbinary, contract_One year, contract_Two Year, internet_service_Fiber Optic, and internet_service_None) from booleans to numeric values using "astype('int64')". Before saving my data, I visually inspected the altered columns to ensure the output was correct.

My annotated code can be found under section '**C4: Data Transformation**' in the file titled '**D208 Task 2.ipynb**'.

C5: Data CSV File

My cleaned and transformed data can be found in the csv file titled '**clean_churn_data_2.csv**'.

Part IV: Model Comparison and Analysis

D1: Initial Logistic Regression Model

My initial model was created by setting my dependent variable and independent variables to 'y' and 'X'. I then used "sm.Logit()" to complete a logistic regression and fit it using "fit()". In this first model I had a total of 26 independent variables/features. My Pseudo R-squared was 0.6199 suggesting it was an initial decent fit. My LLR p-value was 0.00 suggesting that the model could be useful for analysis. My code and regression output for this initial model can be found in section '**D1: Initial Linear Regression Model'** in the file named '**D208 Task 2.ipynb**'. An image of the initial model can also be found in section D3 of this paper for easier comparisons between the initial and reduced model.

D2: Model Reduction Justification

I began my reduction by completing backward eliminations of features with p-values greater than 0.05. I did this because I wanted my regression model to work with features that show statistical significance and likely had a significant relationship with my dependent variable. In each reduction, I removed the least significant feature (the one with the highest p-value above 0.05) and then checked for the next feature until every p-value in my model met the criteria. After completing my backward eliminations, I checked the variance inflation factor (VIF) of my remaining features. My initial logistic model did not provide a multicollinearity warning like my previous linear regression in task 1, so I wanted to see what I was working with. I tested my

features against a VIF threshold of 5 to catch cases of high multicollinearity and avoid worsening the fit if unnecessary. The column monthly_charge had a VIF of about 7.6. This score implies multicollinearity meaning this feature has high correlation with the other independent variables used in our analysis. Keeping this feature could reduce the effect of the model, so I decided to drop it as well. I then rechecked for multicollinearity and all of my selected features now had a VIF below 5. This reduction process left me with 12 final features to be found in my reduced model.

D3: Reduced Logistic Regression Model

After reducing my model, I refit it using "sm.Logit()" and "fit()" again. My final reduced model had 12 total features (not including the constant). The features–techie, phone_service, multiple_lines, tech_support, streaming_tv, streaming_movies, tenure, gender_Male, contract_One year, contract_Two Year, internet_service_Fiber Optic, internet_service_None–all fit my reduction criteria by having VIF values below 5 and p-values below 0.05. My LLR p-value was 0.00 and my Pseudo R-squared was 0.6004.

Here is a photo of my initial model results:

[533]:

Logit Regression Results

Dep. Variable:	churn	**No. Observations:**	10000
Model:	Logit	**Df Residuals:**	9973
Method:	MLE	**Df Model:**	26
Date:	Wed, 30 Oct 2024	**Pseudo R-squ.:**	0.6199
Time:	10:01:13	**Log-Likelihood:**	-2197.6
converged:	True	**LL-Null:**	-5782.2
Covariance Type:	nonrobust	**LLR p-value:**	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
age	0.0022	0.003	0.798	0.425	-0.003	0.008
income	6.426e-07	1.36e-06	0.472	0.637	-2.03e-06	3.31e-06
outage_sec_perweek	-0.0023	0.013	-0.180	0.858	-0.028	0.023
tech_support_contacts	0.0550	0.039	1.427	0.154	-0.021	0.131
yearly_equip_failure	-0.0312	0.061	-0.514	0.607	-0.150	0.088
techie	1.0911	0.102	10.717	0.000	0.892	1.291
port_modem	0.1330	0.077	1.732	0.083	-0.018	0.284
phone_service	-0.2847	0.131	-2.171	0.030	-0.542	-0.028
multiple_lines	0.3591	0.172	2.086	0.037	0.022	0.697
online_security	-0.2703	0.090	-2.993	0.003	-0.447	-0.093
online_backup	-0.1079	0.130	-0.827	0.408	-0.364	0.148
device_protection	-0.0896	0.101	-0.888	0.374	-0.287	0.108
tech_support	-0.2196	0.101	-2.172	0.030	-0.418	-0.021
streaming_tv	1.1334	0.228	4.976	0.000	0.687	1.580
streaming_movies	1.2980	0.264	4.916	0.000	0.781	1.815
tenure	-0.1546	0.048	-3.234	0.001	-0.248	-0.061
monthly_charge	0.0387	0.005	7.688	0.000	0.029	0.049
bandwidth_gb_year	0.0005	0.001	0.833	0.405	-0.001	0.002
area_type_Suburban	-0.0561	0.095	-0.592	0.554	-0.242	0.130
area_type_Urban	0.0394	0.094	0.421	0.674	-0.144	0.223
gender_Male	0.2394	0.086	2.787	0.005	0.071	0.408
gender_Nonbinary	-0.0841	0.261	-0.322	0.748	-0.596	0.428
contract_One year	-3.3793	0.128	-26.466	0.000	-3.630	-3.129
contract_Two Year	-3.4564	0.125	-27.683	0.000	-3.701	-3.212
internet_service_Fiber Optic	-1.9376	0.306	-6.337	0.000	-2.537	-1.338
internet_service_None	-0.7361	0.250	-2.949	0.003	-1.225	-0.247
const	-4.9199	0.510	-9.650	0.000	-5.919	-3.921

Here is a picture of my reduced model results:

```
Optimization terminated successfully.
         Current function value: 0.231042
         Iterations 8
```

[571]:

Logit Regression Results

Dep. Variable:	churn	**No. Observations:**	10000
Model:	Logit	**Df Residuals:**	9987
Method:	MLE	**Df Model:**	12
Date:	Wed, 30 Oct 2024	**Pseudo R-squ.:**	0.6004
Time:	19:28:36	**Log-Likelihood:**	-2310.4
converged:	True	**LL-Null:**	-5782.2
Covariance Type:	nonrobust	**LLR p-value:**	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
techie	1.0807	0.100	10.849	0.000	0.885	1.276
phone_service	-0.3225	0.128	-2.510	0.012	-0.574	-0.071
multiple_lines	1.5780	0.081	19.492	0.000	1.419	1.737
tech_support	0.2443	0.077	3.169	0.002	0.093	0.395
streaming_tv	2.8434	0.093	30.415	0.000	2.660	3.027
streaming_movies	3.3614	0.099	33.982	0.000	3.168	3.555
tenure	-0.1056	0.003	-40.716	0.000	-0.111	-0.101
gender_Male	0.2501	0.075	3.333	0.001	0.103	0.397
contract_One year	-3.1888	0.120	-26.529	0.000	-3.424	-2.953
contract_Two Year	-3.2738	0.117	-27.978	0.000	-3.503	-3.044
internet_service_Fiber Optic	-1.3005	0.088	-14.724	0.000	-1.474	-1.127
internet_service_None	-1.3657	0.107	-12.745	0.000	-1.576	-1.156
const	-0.8702	0.164	-5.305	0.000	-1.192	-0.549

E1: Comparing Initial & Reduced Model

Going from the initial model to the reduced model there was a loss of 14 features (not including the constant). Although the Pseudo R-squared decreased from 0.6199 in the initial model to 0.6004 in the reduced model, the **LLR p-value remained at zero (0.000)** implying that the reduced model is still useful for our analysis as it is below the 0.05 threshold. While I cannot say for certain why the reduced model has a lower Pseudo R-squared than the initial model, the fit of the reduced model may have been impacted by the removal of the monthly_charge feature. While monthly_charge did have multicollinearity issues it was also statistically significant. I explained above (D2) why I felt certain model reduction decisions were necessary. So, while the goodness of fit seems slightly better in the initial model (as implied by the Pseudo R-squared), the accuracy of the model may have been compromised, leading to a misinterpretation of the regression results, had I not mitigated the aforementioned issues.

E2: Analysis Output & Calculations

I first split my data from my reduced model into training and testing groups and then evaluated the model accuracy by printing out my accuracy score and confusion matrix.

E2: Analysis Output & Calculations

[630]:
```
# Splitting my data into training and testing groups
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
split_model = LogisticRegression(solver='lbfgs', max_iter=1000)
split_model.fit(X_train, y_train)

y_test_pred = split_model.predict(X_test)

# Evaluating model accuracy
accuracy_calc = accuracy_score(y_test, y_test_pred)
reduced_cm = confusion_matrix(y_test, y_test_pred)
print(f' Reduced Model Accuracy: {accuracy_calc}')
print('Confusion Matrix: \n', reduced_cm)
```

```
 Reduced Model Accuracy: 0.8933333333333333
Confusion Matrix:
 [[2026  130]
 [ 190  654]]
```

My **accuracy score** for the reduced model test group was **0.8933333333333333** or ~89.3%. My **confusion matrix** indicates that there were 2026 true positives (TP), 130 false negatives (FN), 190 false positives (FP), and 654 true negatives (TN). The total correct predictions (in the top left and bottom right of the matrix) consists of 2680 non-spam samples compared to the 320 incorrect/spam samples (top right and bottom left). The matrix and accuracy score suggest that the model performs well.

E3: Executable Code (error-free)

My entire executable code can also be found in the file titled '**D208 Task 2.ipynb**'.

Part V: Data Summary and Implications

F1: Analysis Results

The equation for the logistic regression is:

ln(p/1-p) = **-0.8702** + 1.0807(**techie**) - 0.3225(**phone_service**) + 1.5780(**multiple_lines**) + 0.2443(**tech_support**) + 2.8434(**streaming_tv**) + 3.3614(**streaming_movies**) - 0.1056(**tenure**) + 0.2501(**gender_Male**) - 3.1888(**contract_One year**) - 3.2738(**contract_Two Year**) - 1.3005(**internet_service_Fiber Optic**) - 1.3657(**internet_service_None**)

I used the coefficients in the equation to describe the behavior of my predictor variables in relation to the target variable.

Keeping all other variables constant:

- techie being 1 results in the natural log odds of churning to increase by 1.0807.

- phone service being 1 causes the natural log odds of churning to decrease by 0.3225.
- multiple lines being 1 results in the natural log odds of churning to increase by 1.5780.
- tech support being 1 causes the natural log odds of churning to increase by 0.2443.
- streaming tv being 1 results in the natural log odds of churning to increase by 2.8434.
- streaming movies being 1 causes the natural log odds of churning to increase by 3.3614.
- a one unit change in tenure causes the natural log odds of churning to change by -0.1056.
- gender_Male being 1 results in the natural log odds of churning to increase by 0.2501.
- contract_One year being 1 causes the natural log odds of churning to decrease by 3.1888.
- contract_Two Year being 1 results in the natural log odds of churning to decrease by 3.2738.
- internet_service_Fiber Optic being 1 causes the natural log odds of churning to decrease by 1.3005.
- internet_service_None being 1 results in the natural log odds of churning to decrease by 1.3657.

I used my LLR p-value to assess the statistical significance of my regression. The value = 0.00. Since it was below the threshold of 0.05 I was able to conclude that my model was statistically relevant and usable for my analysis as it is unlikely to be generated randomly. As for the practical significance, the model is accurate approximately 89.3% of the time, so the company should be able to make meaningful use of this model and its predictions.

Limitations:
One limitation I have is in knowing what exactly is considered customer churn. The data dictionary defines the dependent variable (churn) as whether the customer discontinued service within the last month. However, it does not specify what it means to discontinue service, how many services the customer must discontinue to be considered churning, or the exact services each customer discontinued within the last month. This limitation could impact the interpretation of my model predictions as a customer could churn one or many services offered by our company. Essentially, a case where a customer continues their streaming services but discontinues their phone service to go to another company should not be interpreted the same way as a customer completely discontinuing all of their services with our organization. Knowing more details about the churn conditions can help make this model more useful.

F2: Next Course of Action
The independent variable coefficient estimates with the largest impact on customer churn were **streaming_movies** (3.3614), **contract_Two Year** (-3.2738), **contract_One year** (-3.1888), and **streaming_tv** (2.8434). Customers with two year or one year contracts are significantly less likely to leave compared to the reference variable (contract_Month-to-month). To mitigate churn, the company can use this information and put more promotion efforts into their longer term

contracts–or even decide on the efficacy of offering a month-to-month contract in a future analysis. The company movie and tv streaming services seem to have the opposite effect, where customers who are subscribed to these services are more likely to discontinue. While I have no further details on what the streaming services offer, it could be meaningful to analyze customer satisfaction in regard to these streaming add-ons. A future analysis on qualitative and quantitative customer data related to the customer demographic, subscription fee price and contract length, available content, and/or streaming habits can give the organization more insight on how to improve the relationship between their streaming services and customer retention. While other independent variables like multiple_lines, internet_service (Fiber Optic or None), and techie have a slightly smaller influence, they still have a statistically significant impact on the dependent variable. In similar ways, the company can use this information to mitigate customer churn. They can reassess their target market and look into emphasizing promotions toward tech-savvy customers or families and groups who are looking to connect multiple lines under one plan. They can also offer bundle packages or deals for their internet or streaming services with independent variables that are associated with lower customer churn rates to mitigate the impact of those services.

I would also suggest expanding on the definition of customer churn to provide more details that will aid in the analysis process. Knowing if churn is accounted for by the discontinuation of a single service or if it only counts if the customer leaves the organization completely can enable analysts to provide more relevant and specific insights to combat churn.

Part VI: Demonstration

G: Code Web Sources

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

H: In-Text Citations

Li, S. (2019, February 27). *Building a logistic regression in Python, step by step*. Medium. https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8