D208 – Predictive Modeling (Task 2)

Morrell J. Parrish

Western Governors University

Table of Contents

D208 – Predictive Modeling (Task 2)

A. Research Question

During this course of research, we will determine which variables are indicators or indicates which customers are more likely to "churn" or terminate their services?

A2. Objective or Goals

The objective of this analysis is to perform logistic regression on our churn dataset and define which variable(s) within our churn dataset can be indicators for churn. "The churn rate, also known as the rate of attrition or customer churn; is the frequency in which consumers discontinue doing business with a company. It is commonly represented as the percentage of service subscribers who cancel their memberships within a specified time frame" (Frankenfield, 2022). Defining which variables are indicators businesses can focus on improving in those areas and decrease their churn rate.

B. Assumptions Summary

"A logistic regression model is used to estimate the relationship between a dependent variable and one or more independent variables, but it is used to make a prediction about a categorical variable versus a continuous one" (Lawton, 2022). A logistic regression model makes the following assumptions:

- The response variable is ***binary*** - *(takes on 2 possible outcomes)*
- The observations are independent
- There is no multicollinearity among the explanatory variables
- There are no extreme outliners
- There is a linear relationship between explanatory variables and the logit of the response variable
- The sample size is sufficiently large

B2. Benefits Of Chosen Analytical Tool(s)

The chosen analytical tool for this analysis will be *Python (PyCharm)*. Both *Python* and *R* have strengths and weaknesses; the dataset used in this analysis contains 10000 observations and 50 variables. Both *R* and *Python* have packages/libraries which allow you to cleanse, manage, transform, and perform analysis and statistics. Another reason we will be using *Python* is because it's simple and has a very versatile programming style.

B3. Chosen Technique Explanation

A logistic regression model will allow us to perform regression analysis on a dependent variable that has binary characteristics, and in this case that would be our dependent variable "churn"; also, a logistic regression model will allow us to add or remove variables, by doing this this will help determine if they have a positive or negative impact on predicting if a customer churns or stays.

C. Data Preparation Description

To use the churn dataset in our analysis we will first need to prepare the data. The following steps were taken to prepare the dataset for analysis:

- download the churn dataset
- determine which variables will be used in the analysis
- import the dataset into *PyCharm*
- remove independent variables, demographics, and personal identification variables not being used in the analysis
    - caseorder, customer_id, interaction, UID, city, state, county, zip, lat, lng, population, timezone, job, email, contacts
- determine if any outliners exist and remove them

C2. Summary of Statistics

There are 23 continuous variables, 27 categorical variables and 10000 observations; however, upon further review of the statistical summary some abnormalities can be noted in the following areas children, outage sec perweek, yearly equip failure, tenure, income, monthly charge, and bandwidth gb year. For the purpose of this study we will not use all 50 variables; the *(18)* categorical variables we will use during this analysis are: area, marital, gender, churn, techie, contract, portmodem, tablet, internetservice, phone, multiple, onlinesecurity, onlinebackup, deviceprotection, techsupport, streamingtv, streamingmovies, paperlessbilling, paymentmethod; the *(16)* continuous variables are children, age, income, outage sec per week, yearly equipment failure, tenure, monthly charge, and bandwidth gb year, item1 (timelyresponse), item2 (fixes), item3 (replacements), item4 (reliability), item5 (options), item6 (respectfulness), item7 (courteous), and item8 (listening). The below figure is the summary of statistics for the continuous variables.

	Count	Mean	STD	Min	25%	50%	75%	Max
Children	10000	2.0877	2.147200446	0	0	1	3	10
Age	10000	53.0784	20.69888156	18	35	53	71	89
Income	10000	39806.92677	28199.9167	348.67	19224.7175	33170.605	53246.17	258900.7
Outage_sec_perweek	10000	10.00184816	2.976019188	0.09974694	8.018214	10.01856	11.969485	21.20723
Yearly_equip_failure	10000	0.398	0.635953177	0	0	0	1	6
Tenure	10000	34.52618809	26.44306263	1.00025934	7.917693592	35.430507	61.479795	71.99928
MonthlyCharge	10000	172.6248162	42.94309411	79.97886	139.979239	167.4847	200.734725	290.160419
Bandwidth_GB_Year	10000	3392.34155	2185.294852	155.5067148	1236.470827	3279.536903	5586.14137	7158.98153
TimelyResponse	10000	3.4908	1.037797216	1	3	3	4	7
Fixes	10000	3.5051	1.034640536	1	3	4	4	7
Replacements	10000	3.487	1.027976981	1	3	3	4	8
Reliability	10000	3.4975	1.025816251	1	3	3	4	7
Options	10000	3.4929	1.024819309	1	3	3	4	7
Respectfulness	10000	3.4973	1.033585768	1	3	3	4	8
Courteous	10000	3.5095	1.028501595	1	3	4	4	7
Listening	10000	3.4956	1.028633292	1	3	3	4	8

Figure 1: Summary of Statistics

C3. Data Preparation Steps

To use the churn dataset in our analysis we will first need to prepare the data:

- import the dataset into *Python (PyCharm)*

- view the dataframe's description, structure, and data types
- view summary statistics
- evaluate the dataset, remove null or missing values
- remove any outliners
- remove demographics, and personal identification
    - caseorder, customer_id, interaction, UID, city, state, county, zip, lat, lng, population, area, timezone, job, email, contacts
- convert binomial variables (yes/no to 1 and 0) to numerical variables
- view univariate and bivariate visuals

The following code below was used to prepare our data:

```
# Load data set into Pandas dataframe
df = pd.read_csv('churn_clean.csv')

# Remove less meaningful demographic variables
df = df.drop(columns=['CaseOrder', 'Customer_id', 'Interaction', 'UID',
'City', 'State', 'County', 'Zip', 'Lat', 'Lng', 'Population', 'TimeZone',
'Email', 'Contacts', 'Job'])

# Display Churn dataframe
print(df)

# Rename last 8 columns
df.rename(columns={'Item1': 'TimelyResponse', 'Item2': 'Fixes', 'Item3':
'Replacements', 'Item4': 'Reliability','Item5': 'Options', 'Item6':
'Respectfulness', 'Item7': 'Courteous', 'Item8': 'Listening'}, inplace=True)

# Get column info
print(df.info())

# Describe Churn dataset
print(df.describe())

# Save stats summary to excel
df.describe().to_excel('summary_stat.xlsx', index=False)

# Create Seaborn boxplots for continuous variables
fig2, axs = plt.subplots(4, 2, figsize=(9, 9))

sns.boxplot(y='Churn', x='Children', data=df, color="orange", ax=axs[0, 0])
sns.boxplot(y='Churn', x='Age', data=df, color="gold", ax=axs[0, 1])
sns.boxplot(y='Churn', x='Outage_sec_perweek', data=df, color="turquoise",
ax=axs[1, 0])
```

```
sns.boxplot(y='Churn', x='Yearly_equip_failure', data=df, color="red",
ax=axs[1, 1])
sns.boxplot(y='Churn', x='Tenure', data=df, color="pink", ax=axs[2, 0])
sns.boxplot(y='Churn', x='Income', data=df, color="silver", ax=axs[2, 1])
sns.boxplot(y='Churn', x='MonthlyCharge', data=df, color="green", ax=axs[3,
0])
sns.boxplot(y='Churn', x='Bandwidth_GB_Year', data=df, color="darkblue",
ax=axs[3, 1])

plt.tight_layout(pad=0.4, w_pad=0.5, h_pad=1.0)

plt.show()

# Create Churn Visualizations with categorical variables

df3 = df[['Area', 'Gender', 'Techie', 'Port_modem', 'OnlineSecurity',
'OnlineBackup', 'Marital', 'Contract', 'Tablet', 'InternetService',
'TechSupport', 'PaperlessBilling', 'StreamingTV', 'Phone', 'Multiple',
'StreamingMovies', 'PaymentMethod', 'Churn']]

for index, category in enumerate(df3):
    plt.subplots(1, 1, figsize=(6, 6))

    order = sorted(df3[category].unique())
    ax = sns.countplot(category, data=df3, hue='Churn', order=order)
    ax.set_ylabel('')

    bars = ax.patches
    half = int(len(bars) / 2)
    left_bars = bars[:half]
    right_bars = bars[half:]

    for left, right in zip(left_bars, right_bars):
        height_l = left.get_height()
        height_r = right.get_height()
        total = height_l + height_r

        ax.text(left.get_x() + left.get_width() / 2., height_l + 40,
'{0:.0%}'.format(height_l / total), ha="center")
        ax.text(right.get_x() + right.get_width() / 2., height_r + 40,
'{0:.0%}'.format(height_r / total), ha="center")

plt.show()

# Convert binary variables (yes/no, female/male) to 0 or 1
df['DmyGender'] = [1 if v == 'Male' else 0 for v in df['Gender']]
df['DmyChurn'] = [1 if v == 'Yes' else 0 for v in df['Churn']]
df['DmyTechie'] = [1 if v == 'Yes' else 0 for v in df['Techie']]
df['DmyContract'] = [1 if v == 'Two Year' else 0 for v in df['Contract']]
df['DmyPort_modem'] = [1 if v == 'Yes' else 0 for v in df['Port_modem']]
df['DmyTablet'] = [1 if v == 'Yes' else 0 for v in df['Tablet']]
df['DmyInternetService'] = [1 if v == 'Fiber Optic' else 0 for v in
df['InternetService']]
df['DmyPhone'] = [1 if v == 'Yes' else 0 for v in df['Phone']]
df['DmyMultiple'] = [1 if v == 'Yes' else 0 for v in df['Multiple']]
df['DmyOnlineSecurity'] = [1 if v == 'Yes' else 0 for v in df['OnlineSecurity']]
df['DmyOnlineBackup'] = [1 if v == 'Yes' else 0 for v in df['OnlineBackup']]
df['DmyDeviceProtection'] = [1 if v == 'Yes' else 0 for v in df['DeviceProtection']]
```

```
df['DmyTechSupport'] = [1 if v == 'Yes' else 0 for v in df['TechSupport']]
df['DmyStreamingTV'] = [1 if v == 'Yes' else 0 for v in df['StreamingTV']]
df['DmyStreamingMovies'] = [1 if v == 'Yes' else 0 for v in df['DmyStreamingMovies']]
df['DmyPaperlessBilling'] = [1 if v == 'Yes' else 0 for v in df['PaperlessBilling']]

# Drop original categories
df4 = df.drop(columns=['Gender', 'Churn', 'Techie', 'Contract', 'Port_modem',
'Tablet', 'InternetService', 'Phone', 'Multiple', 'OnlineSecurity',
'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV',
'StreamingMovies', 'PaperlessBilling'])

print(df4.describe())
```

C4. Univariate and Bivariate Visualizations

```
# Create Seaborn boxplots for continuous variables

fig2, axs = plt.subplots(4, 2, figsize=(9, 9))

sns.boxplot(y='Churn', x='Children', data=df, color="orange", ax=axs[0, 0])
sns.boxplot(y='Churn', x='Age', data=df, color="gold", ax=axs[0, 1])
sns.boxplot(y='Churn', x='Outage_sec_perweek', data=df, color="turquoise", ax=axs[1, 0])
sns.boxplot(y='Churn', x='Yearly_equip_failure', data=df, color="red", ax=axs[1, 1])
sns.boxplot(y='Churn', x='Tenure', data=df, color="pink", ax=axs[2, 0])
sns.boxplot(y='Churn', x='Income', data=df, color="silver", ax=axs[2, 1])
sns.boxplot(y='Churn', x='MonthlyCharge', data=df, color="green", ax=axs[3, 0])
sns.boxplot(y='Churn', x='Bandwidth_GB_Year', data=df, color="darkblue", ax=axs[3, 1])

plt.tight_layout(pad=0.4, w_pad=0.5, h_pad=1.0)

plt.show()
```

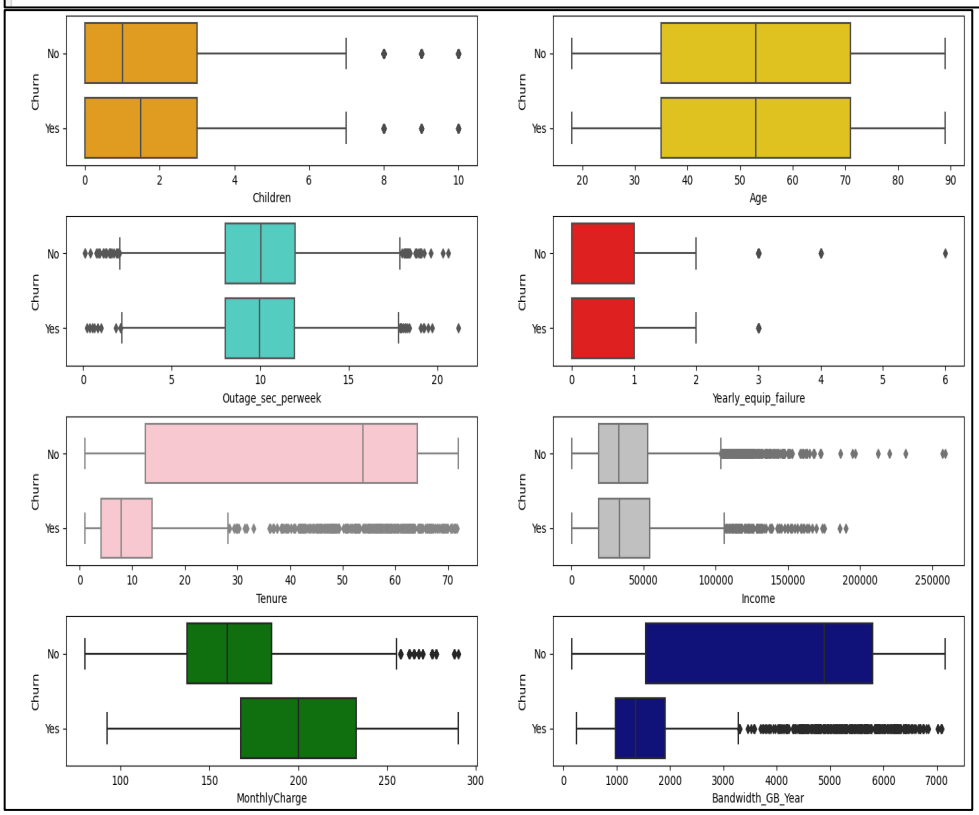
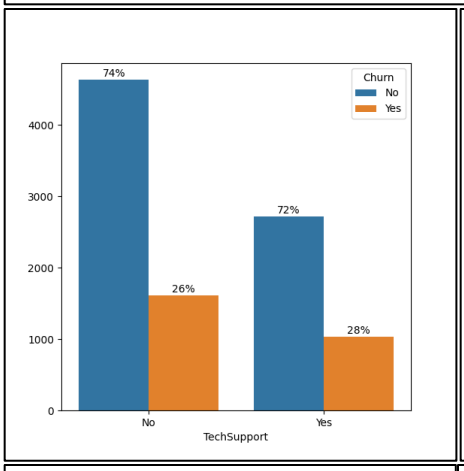
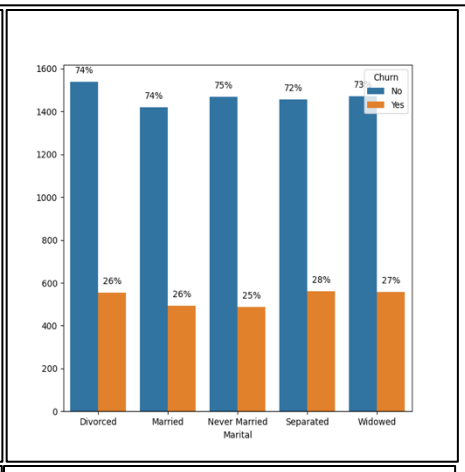
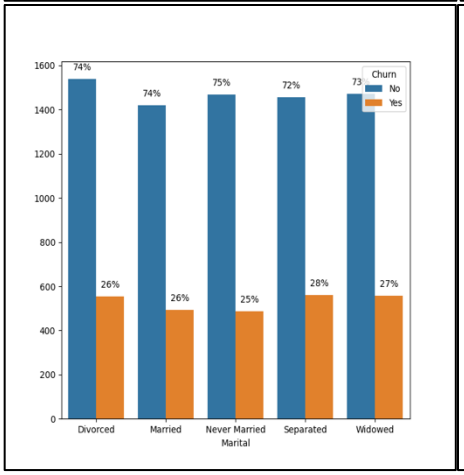


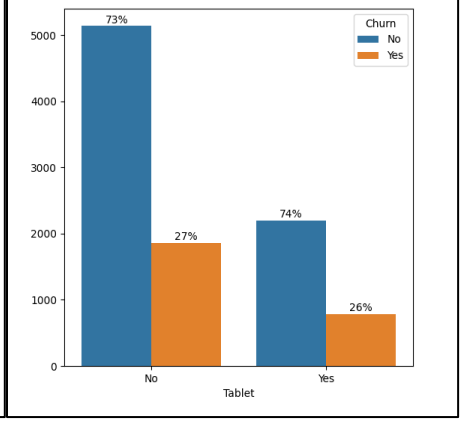
Figure 2: Boxplots for Continuous Variables

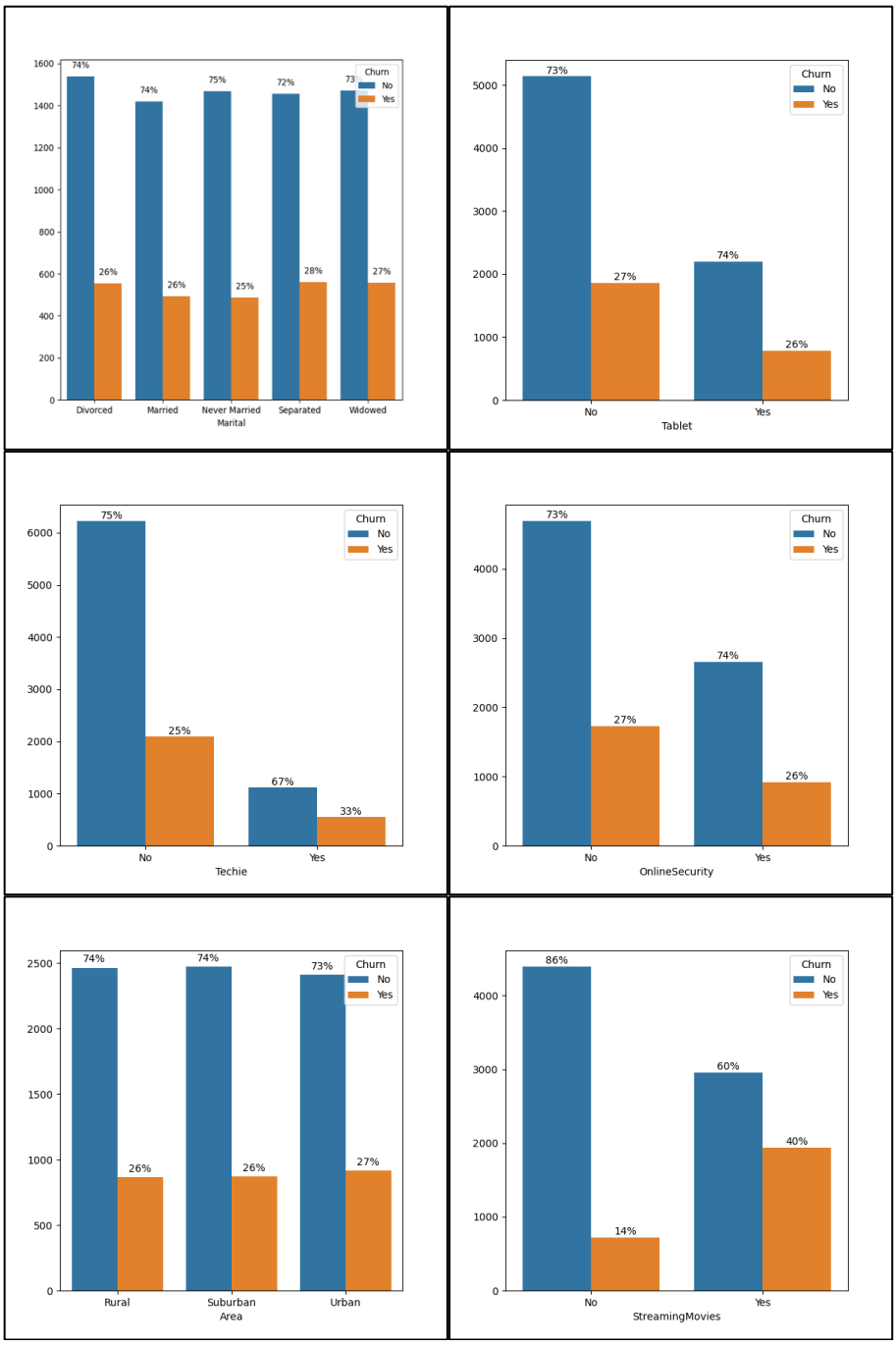
```
# Create Churn Visualizations

df3 = df[['Area', 'Gender', 'Techie', 'Port_modem', 'OnlineSecurity', 'OnlineBackup', 'Marital',
          'Contract', 'Tablet', 'InternetService', 'TechSupport', 'PaperlessBilling', 'StreamingTV',
          'Phone', 'Multiple', 'StreamingMovies', 'PaymentMethod', 'Churn']]

for index, category in enumerate(df3):
    plt.subplots(1, 1, figsize=(6, 6))

    order = sorted(df3[category].unique())
    ax = sns.countplot(category, data=df3, hue='Churn', order=order)
    ax.set_ylabel('')

    bars = ax.patches
    half = int(len(bars) / 2)
    left_bars = bars[:half]
    right_bars = bars[half:]

    for left, right in zip(left_bars, right_bars):
        height_l = left.get_height()
        height_r = right.get_height()
        total = height_l + height_r

        ax.text(left.get_x() + left.get_width() / 2., height_l + 40, '{0:.0%}'.format(height_l / total), ha="center")
        ax.text(right.get_x() + right.get_width() / 2., height_r + 40, '{0:.0%}'.format(height_r / total), ha="center")

plt.show()
```

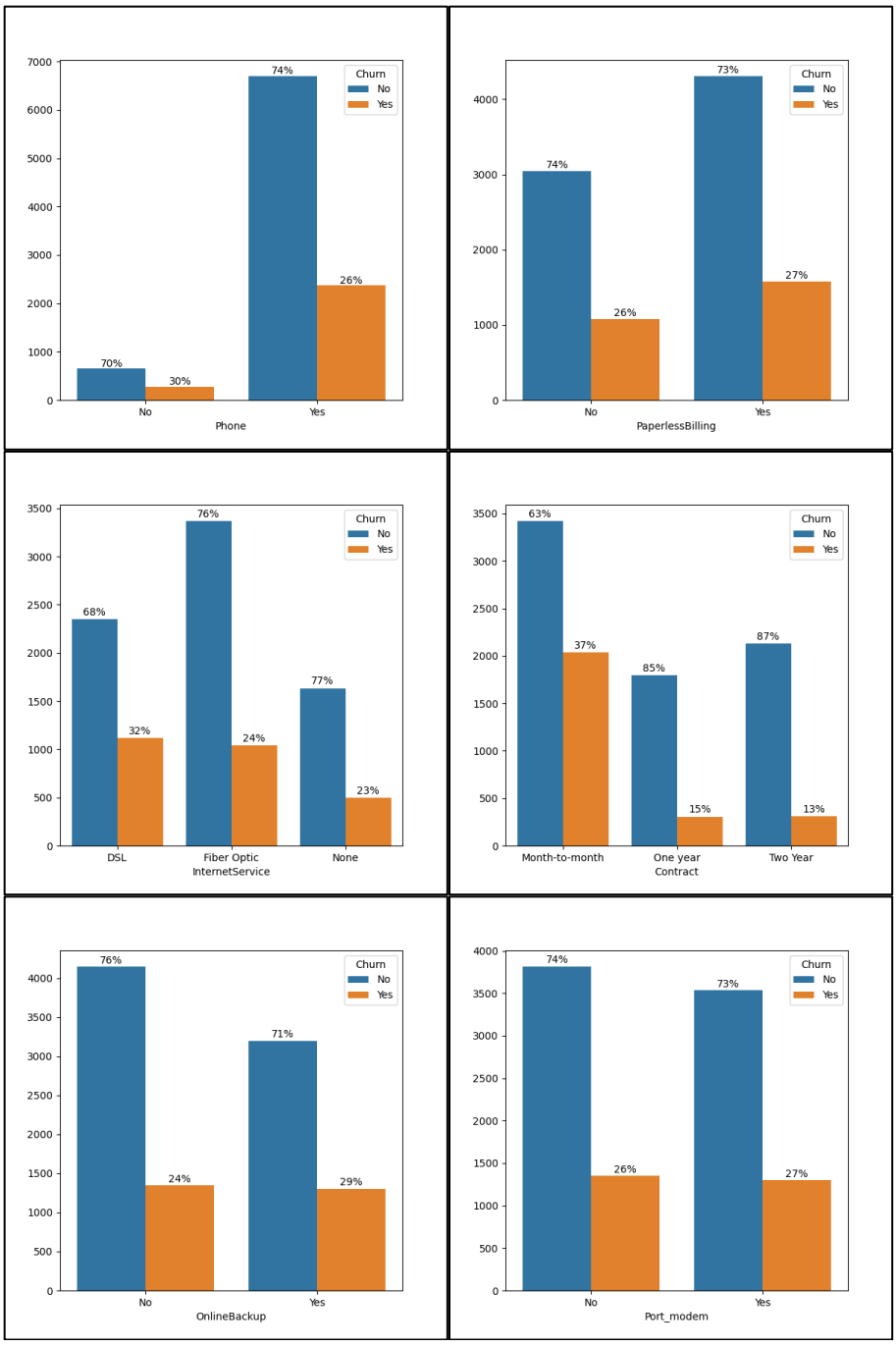









Churn
No
Yes
74%
26%
74%
26%
75%
25%
72%
28%
73%
27%
Divorced
Married
Never Married
Separated
Widowed
Marital
73%
27%
74%
26%
No
Yes
Tablet
75%
25%
67%
33%
No
Yes
Techie
73%
27%
74%
26%
No
Yes
OnlineSecurity
74%
26%
74%
26%
73%
27%
Rural
Suburban
Urban
Area
86%
14%
60%
40%
No
Yes
StreamingMovies


Churn
No
Yes
70%
30%
74%
26%
No
Yes
Phone
74%
26%
73%
27%
No
Yes
PaperlessBilling
68%
32%
76%
24%
77%
23%
DSL
Fiber Optic
None
InternetService
63%
37%
85%
15%
87%
13%
Month-to-month
One year
Two Year
Contract
76%
24%
71%
29%
No
Yes
OnlineBackup
74%
26%
73%
27%
No
Yes
Port_modem

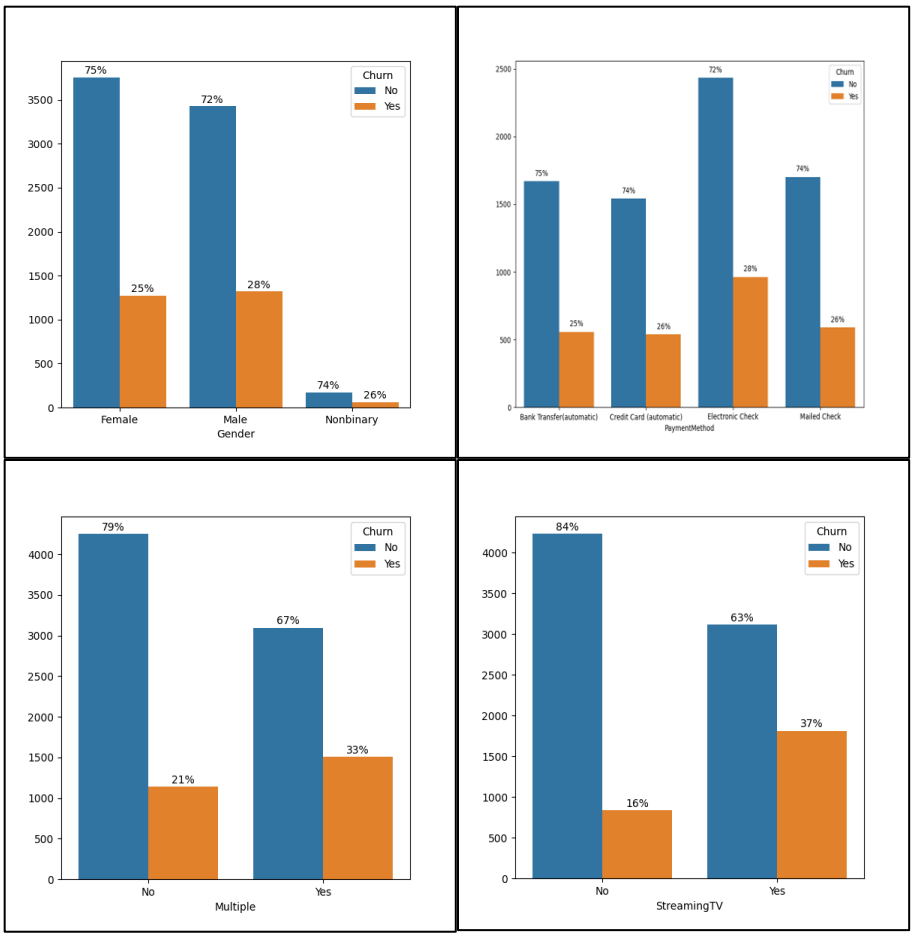


Figure 3: Bar-plots for Categorical Variables

C5. Churn Data Set

The prepared dataset used for this analysis has been uploaded with the assessment file.

D. Model Comparison and Analysis

With the variables identified in C2 we will create our initial logistic regression model.

```
# Split the dataset into a training and testing set. Using an 80/20 testing/training split

x_train, x_test, y_train, y_test = train_test_split(df5.drop('Churn', axis=1), df5['Churn'],
                                                    test_size=0.2, random_state=200)

# Model evaluation default parameters
LogReg = LogisticRegression(solver='liblinear')
res = LogReg.fit(x_train, y_train)

# Classification report precision
y_pred = LogReg.predict(x_test)

# Print classification report
print(classification_report(y_test, y_pred))

# Print predictions
print(y_pred)
```

```
              precision    recall  f1-score   support

           0       0.88      0.92      0.90      1485
           1       0.73      0.64      0.68       515

    accuracy                           0.85      2000
   macro avg       0.80      0.78      0.79      2000
weighted avg       0.84      0.85      0.84      2000
```

Figure 4: Classification Report

According to the classification report for our initial model, 73% of the customers predicted to churn did so. The model also only correctly predicted this outcome for 64% of those customers. Our *f1-score* of 68% indicates that this model does a semi-descent job predicting whether a customer will churn. There were 27 variables evaluated during this logistic regression model; however, not all of the variables are/were needed; we can use a heatmap to determine which variables are most useful.

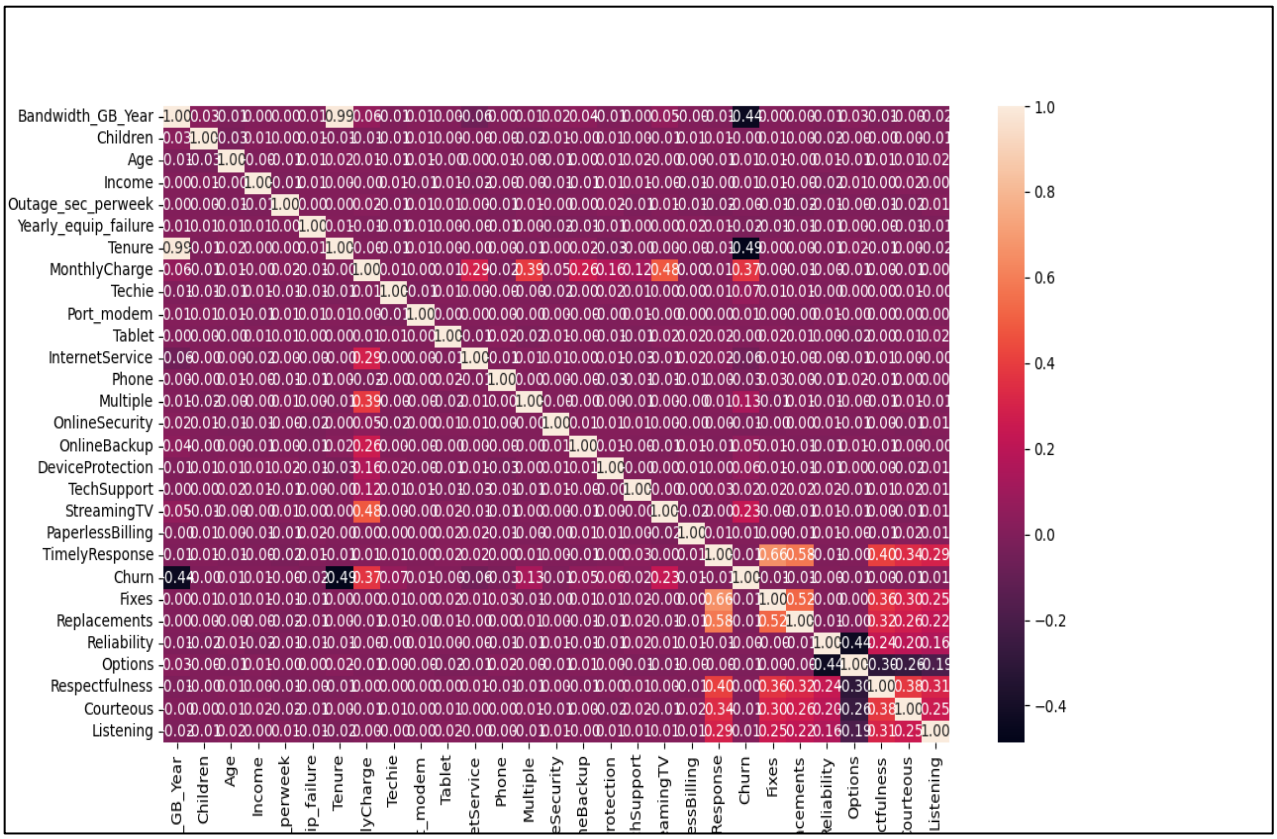


Figure 5: Heatmap with all variables

With all of the variables, the heatmap is difficult to read; however, it should be noted that there is some correlation between tenure and bandwidth gb year. Because our heatmap is difficult to interpret visually, we will look at the logit summary to get a better picture of our variables.

Logit Regression Results

Dep. Variable:	Churn	No. Observations:	8000
Model:	Logit	Df Residuals:	7972
Method:	MLE	Df Model:	27
Date:	Sun, 03 Jul 2022	Pseudo R-squ.:	0.4554
Time:	14:37:29	Log-Likelihood:	-2527.3
converged:	True	LL-Null:	-4641.0
Covariance Type:	nonrobust	LLR p-value:	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
Bandwidth_GB_Year	0.0008	0.000	3.174	0.002	0.000	0.001
Children	-0.0505	0.018	-2.743	0.006	-0.087	-0.014
Age	0.0011	0.002	0.564	0.572	-0.003	0.005
Income	-1.425e-06	1.26e-06	-1.134	0.257	-3.89e-06	1.04e-06
Outage_sec_perweek	-0.0558	0.011	-4.926	0.000	-0.078	-0.034
Yearly_equip_failure	-0.0726	0.056	-1.292	0.196	-0.183	0.038
Tenure	-0.1513	0.022	-7.007	0.000	-0.194	-0.109
MonthlyCharge	0.0361	0.002	19.613	0.000	0.032	0.040
Techie	0.6868	0.092	7.485	0.000	0.507	0.867
Port_modem	0.0120	0.071	0.170	0.865	-0.126	0.150
Tablet	-0.1667	0.077	-2.154	0.031	-0.318	-0.015
InternetService	-1.3496	0.130	-10.368	0.000	-1.605	-1.094
Phone	-0.6305	0.112	-5.624	0.000	-0.850	-0.411
Multiple	-0.2286	0.082	-2.779	0.005	-0.390	-0.067
OnlineSecurity	-0.3779	0.077	-4.929	0.000	-0.528	-0.228
OnlineBackup	-0.3940	0.076	-5.178	0.000	-0.543	-0.245
DeviceProtection	-0.3657	0.073	-4.976	0.000	-0.510	-0.222
TechSupport	-0.3358	0.076	-4.414	0.000	-0.485	-0.187
StreamingTV	0.1010	0.086	1.181	0.238	-0.067	0.269
PaperlessBilling	-0.0556	0.071	-0.779	0.436	-0.195	0.084
TimelyResponse	-0.0299	0.050	-0.594	0.553	-0.129	0.069
Fixes	0.0011	0.047	0.023	0.982	-0.092	0.094
Replacements	-0.0697	0.043	-1.627	0.104	-0.154	0.014
Reliability	-0.2555	0.035	-7.201	0.000	-0.325	-0.186
Options	-0.3363	0.034	-9.928	0.000	-0.403	-0.270
Respectfulness	-0.1009	0.041	-2.466	0.014	-0.181	-0.021
Courteous	-0.1208	0.039	-3.137	0.002	-0.196	-0.045
Listening	-0.0805	0.037	-2.199	0.028	-0.152	-0.009

Figure 6: Initial Logistic Regression Model Results

We will concentrate on the p-values; p-values less than 0.05 are a good candidate for further research; below is the logit regression results. After review our initial logistic regression model summary we will redice our model down to the following variables: bandwidth gb year, age, income, tenure, yearly equip failure, port modem, tablet, streaming tv, paperless billing, timely response, churn, fixes, and replacements.

D2. Justification of Based Variable Selection Procedure and Model Evaluation Metric

Originally our dataset contained 50 variables and 10000 observations: to choose the best variables for the analysis a correlation Matrix was created. After creating the correlation heatmap and comparing the heatmap to other visualizations that were created; it was noted that

some of the features did not have much significance and/or were strongly correlated. After running the first correlation matrix, I then ran the stats models logit method on the same dataset; this allowed me to take a closer look at my variables and their significance level, I removed any variables which had a p-value greater than 0.05. The following variables were removed because they did not have much significance and/or were strongly correlated: age, children, income, yearly_equip_failure, dmyTable, dmyStreamingTV, dmyPort_Modem, dmyPaperessBilling', timelyresponse, fixes, dmyMultiple, replacements, reliability, bandwidth_gb_year. The following variables will be used in our logistic regression model: outage_sec_perweek, tenure, monthlycharge, dmyTechie, dmyInternetService, dmyPhone, dmyOnlineSecurity, dmyOnlineBackup, dmyDeviceProtection, dmyTechSupport, dmyChurn, options, respectfulness, courteous, and listening.

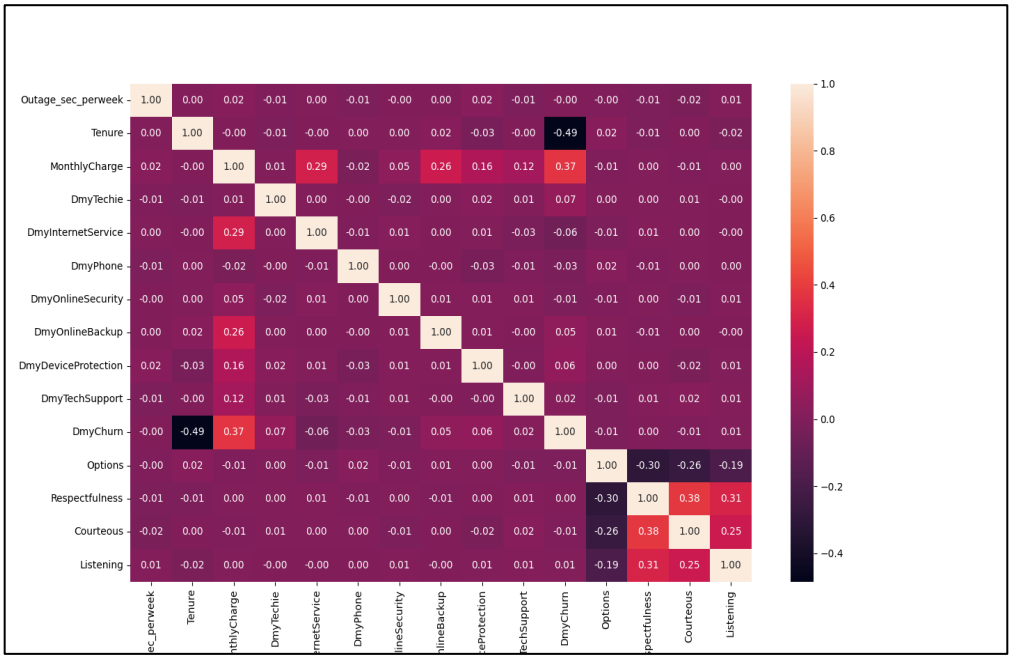


Figure 7: Reduced Logistic Regression Model Heatmap

D3. Logistic Regression Model

```
236  # check for precision, recall and f1 score
237  print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.88	0.93	0.90	1485
1	0.76	0.63	0.69	515
accuracy			0.85	2000
macro avg	0.82	0.78	0.80	2000
weighted avg	0.85	0.85	0.85	2000

Figure 9: Classification Report

Logit Regression Results

Dep. Variable:	DmyChurn	No. Observations:	8000
Model:	Logit	Df Residuals:	7986
Method:	MLE	Df Model:	13
Date:	Sat, 09 Jul 2022	Pseudo R-squ.:	0.4444
Time:	13:58:50	Log-Likelihood:	-2578.8
converged:	True	LL-Null:	-4641.0
Covariance Type:	nonrobust	LLR p-value:	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
Outage_sec_perweek	-0.0745	0.011	-6.775	0.000	-0.096	-0.053
Tenure	-0.0819	0.002	-37.592	0.000	-0.086	-0.078
MonthlyCharge	0.0373	0.001	33.665	0.000	0.035	0.039
DmyTechie	0.6504	0.091	7.151	0.000	0.472	0.829
DmyInternetService	-1.6276	0.081	-20.097	0.000	-1.786	-1.469
DmyPhone	-0.8006	0.109	-7.378	0.000	-1.013	-0.588
DmyOnlineSecurity	-0.3276	0.073	-4.460	0.000	-0.472	-0.184
DmyOnlineBackup	-0.3542	0.073	-4.876	0.000	-0.497	-0.212
DmyDeviceProtection	-0.3539	0.072	-4.949	0.000	-0.494	-0.214
DmyTechSupport	-0.4015	0.073	-5.504	0.000	-0.544	-0.259
Options	-0.3388	0.031	-10.976	0.000	-0.399	-0.278
Respectfulness	-0.1892	0.037	-5.154	0.000	-0.261	-0.117
Courteous	-0.1979	0.036	-5.500	0.000	-0.268	-0.127
Listening	-0.1417	0.035	-4.079	0.000	-0.210	-0.074

Figure 10: Reduced Logistic Model Results

According to the classification report for our reduced model, 76% of the customers predicted to churn did so. The model also only correctly predicted this outcome for 63% of those customers. Our *f1-score* of 69% indicates that this reduced model did improve on predicting whether a customer will churn.

E. Data Set Analyzation

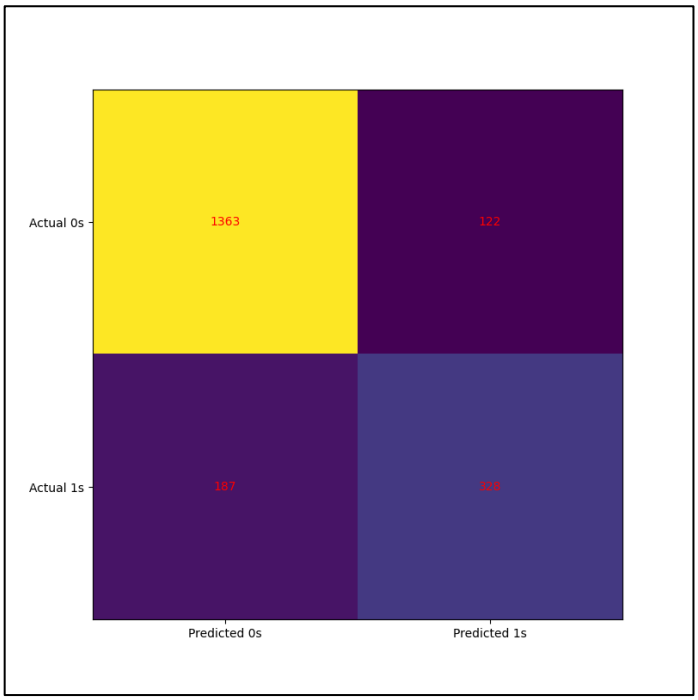


Figure 11: Initial Confusion Matrix

Out of 2000 random observations our initial logistic model produced the following confusion matrix with the following results tn = 1363, fp = 122, fn=187, and tp=328.

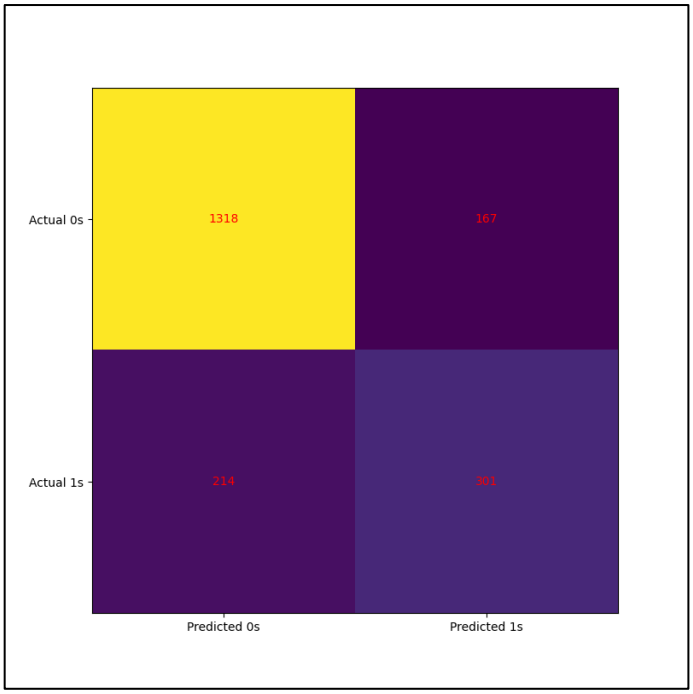


Figure 12: Reduced Model Confusion Matrix

Out of 2000 random observations our reduced logistic model produced the following confusion matrix with the following results tn = 1318, fp = 167, fn=214, and tp=301.

E3. Logistic Regression Model Code

```
# Split the dataset into a training and testing set. Using an 80/20
testing/training split

x_train, x_test, y_train, y_test = train_test_split(df5.drop('Churn',
axis=1), df5['Churn'], test_size=0.2, random_state=200)

# Model evaluation default parameters
LogReg = LogisticRegression(solver='liblinear')
res = LogReg.fit(x_train, y_train)

# Classification report precision
y_pred = LogReg.predict(x_test)

# Print classification report
print(classification_report(y_test, y_pred))

# Print predictions
print(y_pred)

# Building and Printing the Logit Summary

log_reg = sm.Logit(y_train, x_train).fit()

print(log_reg.summary())

# Create reduced model

x_train, x_test, y_train, y_test = train_test_split (df6.drop('Churn',
axis=1), df6['Churn'], test_size=0.2, random_state=200)

# Model evaluation default parameters
LogReg2 = LogisticRegression(solver='liblinear')

# train model
LogReg2.fit (x_train, y_train)

# Classification report precision
y_pred = LogReg2.predict(x_test)
```

F. Summary

After comparing both models the reduced model did improve the accuracy of predicting churn customers; the reduced model for customers who churned precision increased by 3%, the recall value decreased by 1%, and the *f1-score* increased by 1%. The initial logistic regression

model yielded a precision score of 73%, which was accurately predicted 64% of the time. The reduced model yielded a precision score of 76%, which was accurately predicted 63% of the time.

Regression formula:

y = (-0.07)*outage_sec_perweek + (-0.08)*tenure + (0.04)*monthlycharge + (0.65)*dmytechie + (-1.63)*dmyInternetService +(-0.80)*dmyPhone +(-0.33)* dmyOnlineSecurity + (-0.35)*dmyOnlineBackup + (-0.35)*dmyDeviceProtection + (-0.40)*dmyTechSupport + (-0.34)*Options + (-0.19)*Respectfulness + (-0.2)*Courteous + (-014)*Listening

Interpretation of Coefficients:

The coefficients represent a positive and/or negative multiplier; monthlycharge and dmytechie variables represent a positive predictor for churn, while outage sec perweek, tenure, dmyinternetservice, dmyphone, dmyonlinesecurity, dmyonlinebackup, dmydeviceprotection, dmytechsupport, options, respectfulness, courteous and listening represent a negative predicator for churn.

Limitations of Analysis:

There are some limitations to this analysis, this analysis does not cover those customers that signed up during a promotional period and ended their subscription after that period ended; this analysis is also a small representative of a larger dataset and covers a limited period of time.

F2. Recommended Course of Action

Since the reduced model only increased our predictions slightly, I would recommend including some of the omitted variables to see if they improve the results of the model; or we can also reevaluate the variables which were included in our reduced model to see which or if any of them are having a negative impact on our reduced models. Other recommendations are to

include or create incentives for customers to sign longer contracts (1 to 2 years) and include more services; the analysis showed the more services a customer has the less likely they are to churn. I would also recommend looking into the customers that signed up during a promotional period; see why they signed up and why they discontinued their services after the promotional period.

G. Panopto video recording

Video Link

References

Frankenfield, J. (2022, February 8). Churn rate. Investopedia. Retrieved May 13, 2022, from https://www.investopedia.com/terms/c/churnrate.asp

Lawton, G., Burns, E., & Rosencrance, L. (2022, January 20). What is logistic regression? - definition from Searchbusinessanalytics. SearchBusinessAnalytics. Retrieved July 4, 2022, from https://www.techtarget.com/searchbusinessanalytics/definition/logistic-regression

Narkhede, S. (2021, June 15). Understanding Confusion Matrix. Medium. Retrieved July 4, 2022, from https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

NBM2 TASK 2: Logistic Regression For Predictive Modeling. WGU Performance Assessment. (n.d.). Retrieved July 4, 2022, from https://tasks.wgu.edu/student/000194226/course/23300006/task/2788/overview