D208 Logistic Regression for Predictive Modeling Performance Assessment, Task 2

Alexa R. Fisher

Western Governors University

Degree: M.S. Data Analytics

Table of Contents

Part I. Research Question for Data Analysis

A1. Research Question for Analysis

The research question for this thesis was, “What variables contribute to the probability of Churn?”. This thesis was evaluated via the computation of logistic regression on the telecommunications dataset. The target or dependent variable used for this regression model was “Churn”. This attribute described whether a customer discontinued service within the last month. Several explanatory variables were used in the formation of the predictive modeling method. The variables included the following: “Population”, “Area”, “Children”, “Age”, “Income”, “Marital”, “Gender”, “Contacts”, “Yearly_equip_failure”, “Contract”, “InternetService”, “PaymentMethod”, “Tenure”, and “MonthlyCharge”. The predictive modeling technique called the logistic regression model estimated the probability or log odds of the target binary variable occurring given the association between it and the explanatory variables (Broeck, Introduction to Regression with statsmodels in Python, n.d.).

A2. Objective From Analysis

The main objective of this analysis was to identify if specific explanatory variables were indicative of “Churn” within the telecommunication dataset. This goal supplied insightful information about the probability of certain variables changing the likelihood of customer departure. In the telecommunication service dataset, it was observed that customers vary in age, gender, household size, and income. Another list of observations was the type of internet service, the contract terms, as well as how long they have been a client. Lastly, it noted the yearly failure rates, how much a customer paid monthly, and the payment method utilized. The stakeholders may want to investigate if the specified variables have a higher association with “Churn”. This investigation can provide an understanding of various problematic areas to be addressed for retention efforts (Expert Panel, Forbes Agency Council, 2019).

Part II. Method Justification

B1. Summary for Assumptions of Logistic Regression Model

The logistic regression model assumptions can be summarized in the following points.

1. The target variable is binary (Bobbitt, 2020).
2. The sample size of the logistic regression must be large (Statistic Solutions, n.d.).
3. The explanatory variables are Independent (Bobbitt, 2020)
4. There is no multicollinearity (Bobbitt, 2020).
5. There are no extreme outliers (Bobbitt, 2020).

The noted points above were verified in many ways. The count of unique values within the variable could confirm if a variable was binary. The presence of only two unique values noted the variable as binary. Visualizations such as boxplots and histograms confirmed the presence of outliers within the dataset. Independence could be confirmed by checking each variable for any matched values or duplicates. The confirmation of no multicollinearity was determined by calculating the variance inflation factor or VIF score. A VIF score of greater than ten notated a higher chance of multicollinearity. Lastly, logistic regression required the sample size to be greater than ten for each independent variable. The review of the summary statistics confirmed the count of each variable within this dataset.

B2. Benefits of Using Python

The selected programming language, Python, had numerous benefits to support the logistic regression model. It was general-purpose, object-oriented programming language. Python allowed for libraries and packages to be imported for data analysis procedures. There was an ample collection of packages such as Pandas, NumPy, Sklearn, Statsmodels, Matplotlib, and Seaborn to provide computations and visualizations. The use of these allowed for the dataset to be explored and provided intuitive information from which to conclude. Data were explored, cleaned, and analyzed with the use of tools provided by Pandas. The ability to work with arrays during data cleaning, data wrangling, and predictive modeling phases was completed with the use of NumPy. The Matplotlib and Seaborn libraries provided means to visualize data during all stages of analysis. Statistics were computed within the Python environment via Sklearn and Statsmodels. These libraries had numerous toolsets such as variance inflation factor, logit, confusion matrix, and train test split. The variance inflation factor or VIF method was utilized for checking on multicollinearity. The logit regression model showed the coefficients within the dataset compared to the dependent variable as it is associated with the probability of occurrence (IBM, n.d.). The Train Test Split and confusion matrix packages provided data for visualization of the True or False positive and negative predictions. Each step of the analysis procedures such as data cleaning, data exploration, data wrangling, and predictive modeling was able to be completed via these described libraries.

B3. Justification for Logistic Regression Technique

The logistic regression method was justified as an appropriate technique to analyze the presented research question, “What variables contribute to the probability of Churn?”. Logistic regression was described as a generalized linear model that has a binary target variable (Broeck, Introduction to Regression with statsmodels in Python, n.d.). In this telecommunication dataset, “Churn” was selected as the target variable as it had a “Yes or No” binary value. The additional variables presented more insight into the probability of the target variable event occurring (IBM, n.d.). Comparing the independent variables against this target variable revealed the odds ratio of “Churn” being “Yes” in relation to each of the independent variables. This insight would provide knowledge as of which variables would be problematic in terms of retention. The stakeholders can utilize this for targeted solutions and future marketing decisions (Expert Panel, Forbes Agency Council, 2019).

Part III. Data Preparation

C1. Data Preparation: Goals

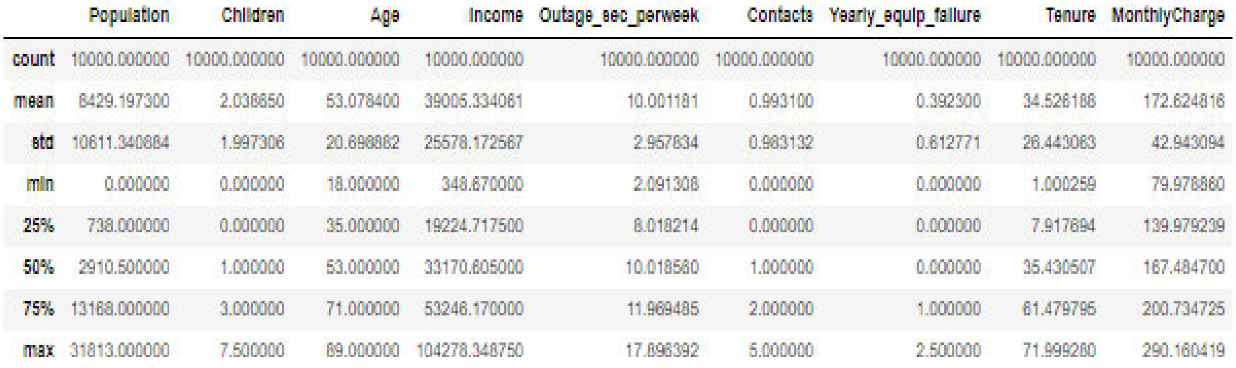
There were numerous data preparation and manipulation goals for the telecommunication dataset to analyze the research question. The main objectives were the data cleaning, data exploration, and data wrangling phases. To complete the data cleaning process, the target and explantory variables had to be free of duplications, null values, and outliers. This was completed by manipulating the data to find these features and resolve them. The features were resolved by capping outliers and removing the unneeded variables from the dataset. Capping of the outliers was utilized by setting the outliers to the upper and lower limits of the variables found via the

IQR method. Completing this method of treatment allowed the distribution of the data to remain the same across all variables. The data exploration goal was solved by summary statistics and visualizations. Visualizations included graphical plots such as boxplots and bar plots. Each of the explanatory variables was explored to show their degree of cardinality as well as their bivariate and univariate statistical findings. The data wrangling goal was accomplished by re-expressing the categorical variables via various encoding methods. One-hot encoding and ordinal encoding were applied to the categorical variables within this telecommunication dataset. The categorical variable conversion to numerical values put the attributes in the ideal format for predictive modeling. The completion of these goals allowed for logistic regression techniques to be used.

C2. Summary of Statistics

One of the key parts of the exploratory analysis of the telecommunication dataset was the summary of statistics. Summary statistics were provided by utilizing the .describe() function in Python. This function specified the count, mean, standard deviation, quantile information, the minimum, and the maximum value for each quantitative variable. The following explanatory variables were quantitative: “Population”, “Children”, “Age”, “Income”, “Outage_sec_perweek”, “Contacts”, “Yearly_equip_failure”, “Tenure”, and “MonthlyCharge”. Count was the total entries found within each variable. The variables’ count for all variables were 10,000 entries. The minimum and maximum are the lowest and highest values respectfully found in each variable. The value provided insight into the range of the represented data. The mean was described as the measure of the location. This was accomplished by averaging the entries. The standard deviation noted the average amount of variability. Variability referenced how far on average a value lies from the mean. The quantile percentages were divided into three percentiles. The lower quantile was the 25% percentile. The median of the variable or 50%

percentile was the same as the middle value in the sorted list. The 75% percentile was the upper quantile.



	Population	Children	Age	Income	Outage_sec_perweek	Contacts	Yearly_equip_failure	Tenure	MonthlyCharge
count	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000
mean	8429.197300	2.038650	53.078400	39005.334061	10.001181	0.993100	0.392300	34.526188	172.624816
std	10611.340884	1.997306	20.698882	25578.172567	2.957834	0.983132	0.612771	26.443063	42.943094
min	0.000000	0.000000	18.000000	348.670000	2.091308	0.000000	0.000000	1.000259	79.978860
25%	738.000000	0.000000	35.000000	19224.717500	8.018214	0.000000	0.000000	7.917694	139.979239
50%	2910.500000	1.000000	53.000000	33170.605000	10.018560	1.000000	0.000000	35.430507	167.484700
75%	13168.000000	3.000000	71.000000	53246.170000	11.969485	2.000000	1.000000	61.479795	200.734725
max	31813.000000	7.500000	89.000000	104278.348750	17.898392	5.000000	2.500000	71.999280	290.160419

The summary statistics were noted within the bivariate analysis from the use of a cross-tabular contingency table. This table also known as a crosstab table provided the bivariate analysis for the target variable, “Churn” against each of the explanatory qualitative variables. The table was plotted on a bar plot to graphically picture the association between the target and explanatory variables such as “Area”, “Marital”, “Gender”, “Contract”, “InternetService”, and “PaymentMethod”.

```
Area   Rural  Suburban  Urban
Churn
No      2464      2473   2413
Yes      863       873    914

Marital  Divorced  Married  Never Married  Separated  Widowed
Churn
No           1539     1418           1468       1454     1471
Yes           553      493            488        560      556

Gender  Female  Male  Nonbinary
Churn
No        3753  3425        172
Yes       1272  1319         59

Contract  Month-to-month  One year  Two Year
Churn
No                  3422      1795      2133
Yes                 2034       307       309
```

```
InternetService   DSL  Fiber Optic  None
Churn
No               2349         3368  1633
Yes              1114         1040   496

PaymentMethod  Bank Transfer(automatic)  Credit Card (automatic)  \
Churn
No                                 1671                     1543
Yes                                 558                      540

PaymentMethod  Electronic Check  Mailed Check
Churn
No                         2435          1701
Yes                         963           589
```

C3. Data Preparation Steps

The data preparation procedures included the following steps on the telecommunication dataset. The steps included three key elements: data cleaning, data exploration, and data wrangling. The phases were applied to the telecommunication dataset before predictive modeling could be performed.

The first step, data cleaning, was utilized to remove duplicates, null values, and outliers. It was best to review all the variable names in the dataset by using the .info() function. It allowed the variables to be visually inspected for any duplicated names. Next, the .duplicated() function checked the full dataset for any duplicated values. The “keep last” provided an additional action to remove any duplicates found by dropping the first one. In the case of this dataset, there were no values dropped or duplicates found.

The .isnull() function was used to find any null or missing values in the telecommunication dataset after the duplicates were resolved. The .sum() function was used in conjunction with the .isnull() function to provide a total count of null values per variable. The telecommunication dataset had no null values found.

The last step in the data-cleaning process was detecting and treating outliers. A created function was made to locate the outliers within this dataset. The function supplied the total amount of outliers per the quantitative variable. This representation provided the notation for the minimum and maximum outlier values as well as the volume of outliers for each of those variables. The function created was called “find_outliers”. It utilized the interquartile range or IQR method. This method calculated the IQR by subtracting the first quantile from the third quantile. It was further separated into two equations to mathematically calculate the lower and upper bounds for each variable. This was shown as follows: Q1 – 1.5 * IQR = lower bound value and Q3 + 1.5 * IQR = upper bound value. In the IQR method, the values found outside the lower bound and upper bound were considered outliers. Another process of detecting outliers was box plotting the variables to check for the distribution and visualizing the outliers.

Once the outliers were identified, they needed to be treated. The outliers were treated via another created function called “find_boundary” that used the same IQR method distinction. After the outlier boundaries were identified, the NumPy .where() function was used to manipulate the values found outside of those limits. This process was called capping, which set the outliers to the closest lower or upper limit accordingly. Once outliers were treated, the box plots were replotted to ensure the outliers were resolved and the distributions did not change.

Below was the annotated code for finding duplicates, null values, and outliers.

```
df.info()
# checking for duplicates
data.duplicated(keep=’last’)

#checking for null values
data.isnull().sum()
```

```
#finding outliers
def find_outliers(df, var):
    q1 = df[var].quantile(0.25)
    q3 = df[var].quantile(0.75)
    IQR = q3 - q1
    lowerbound = q1-(1.5*IQR)
    upperbound = q3+(1.5*IQR)
    outliers = df[var][((df[var] < (lowerbound)) | (df[var] > (upperbound)))]
    return outliers

#running created to function on quantitative variables
outliers = find_outliers(data, 'Population')
print("number of outliers in Population: "+ str(len(outliers)))
print("max outlier value: "+ str(outliers.max()))
print("min outlier value: "+ str(outliers.min()))

outliers = find_outliers(data, 'Children')
print("number of outliers in Children: "+ str(len(outliers)))
print("max outlier value: "+ str(outliers.max()))
print("min outlier value: "+ str(outliers.min()))

outliers = find_outliers(data, 'Age')
print("number of outliers in Age: "+ str(len(outliers)))
print("max outlier value: "+ str(outliers.max()))
print("min outlier value: "+ str(outliers.min()))

outliers = find_outliers(data, 'Income')
print("number of outliers in Income: "+ str(len(outliers)))
print("max outlier value: "+ str(outliers.max()))
print("min outlier value: "+ str(outliers.min()))
```

```
outliers = find_outliers(data, 'Outage_sec_perweek')
print("number of outliers in Outage_sec_perweek: "+ str(len(outliers)))
print("max outlier value: "+ str(outliers.max()))
print("min outlier value: "+ str(outliers.min()))

outliers = find_outliers(data, 'Email')
print("number of outliers in Email: "+ str(len(outliers)))
print("max outlier value: "+ str(outliers.max()))
print("min outlier value: "+ str(outliers.min()))

outliers = find_outliers(data, 'Contacts')
print("number of outliers in Contacts: "+ str(len(outliers)))
print("max outlier value: "+ str(outliers.max()))
print("min outlier value: "+ str(outliers.min()))

outliers = find_outliers(data, 'Yearly_equip_failure')
print("number of outliers in Yearly_equip_failure: "+ str(len(outliers)))
print("max outlier value: "+ str(outliers.max()))
print("min outlier value: "+ str(outliers.min()))

outliers = find_outliers(data, 'Tenure')
print("number of outliers in Tenure: "+ str(len(outliers)))
print("max outlier value: "+ str(outliers.max()))
print("min outlier value: "+ str(outliers.min()))

outliers = find_outliers(data, 'MonthlyCharge')
print("number of outliers in MonthlyCharge: "+ str(len(outliers)))
print("max outlier value: "+ str(outliers.max()))
print("min outlier value: "+ str(outliers.min()))
```

```
outliers = find_outliers(data, 'Bandwidth_GB_Year')
print("number of outliers in Bandwidth_GB_Year: "+ str(len(outliers)))
print("max outlier value: "+ str(outliers.max()))
print("min outlier value: "+ str(outliers.min()))

#boxplotting all variables showing outliers
boxplot=sns.boxplot(x='Population',data=df)
plt.show()
boxplot=sns.boxplot(x='Children',data=df)
plt.show()
boxplot=sns.boxplot(x='Income',data=df)
plt.show()
boxplot=sns.boxplot(x='Outage_sec_perweek',data=df)
plt.show()
boxplot=sns.boxplot(x='Email',data=df)
plt.show()
boxplot=sns.boxplot(x='Contacts',data=df)
plt.show()
boxplot=sns.boxplot(x='Yearly_equip_failure',data=df)
plt.show()

#treating outliers found.
def find_boundary(df, var):
    Q1 = df[var].quantile(0.25)
    Q3 = df[var].quantile(0.75)
    IQR = Q3-Q1
    lower = Q1-(1.5*IQR)
    upper = Q3+(1.5*IQR)
    return lower , upper
```

```
lower_pop, upper_pop = find_boundary(data, 'Population' )
print("Upper limit for population is" , upper_pop)
print("Lower limit for population is" , lower_pop)
data.Population = np.where(data.Population > upper_pop, upper_pop,
                    np.where(data.Population < lower_pop, lower_pop,
                    data.Population))


lower_kid, upper_kid = find_boundary(data, 'Children')
print("Upper limit for children is" , upper_kid)
print("Lower limit for children is" , lower_kid)
data.Children = np.where(data.Children > upper_kid, upper_kid,
                    np.where(data.Children < lower_kid, lower_kid,
                    data.Children))


lower_inc, upper_inc = find_boundary(data, 'Income')
print("Upper limit for Income is" , upper_inc)
print("Lower limit for Income is" , lower_inc)
data.Income = np.where(data.Income > upper_inc, upper_inc,
                np.where(data.Income < lower_inc, lower_inc, data.Income))


lower_osp, upper_osp = find_boundary(data, 'Outage_sec_perweek')
print("Upper limit for Outage_sec_perweek is" , upper_osp)
print("Lower limit for Outage_sec_perweek is" , lower_osp)
data.Outage_sec_perweek = np.where(data.Outage_sec_perweek > upper_osp,
upper_osp,
                              np.where(data.Outage_sec_perweek < lower_osp,
                      lower_osp, data.Outage_sec_perweek))


lower_eml, upper_eml = find_boundary(data, 'Email')
print("Upper limit for email is" , upper_eml)
print("Lower limit for email is" , lower_eml)
```

```
data.Email = np.where(data.Email > upper_eml, upper_eml,
                      np.where(data.Email < lower_eml, lower_eml, data.Email))


lower_contct, upper_contct = find_boundary(data, 'Contacts')

print("Upper limit for contacts is" , upper_contct)

print("Lower limit for contacts is" , lower_contct)

data.Contacts = np.where(data.Contacts > upper_contct, upper_contct,
                 np.where(data.Contacts < lower_contct, lower_contct,
                 data.Contacts))


lower_yef, upper_yef = find_boundary(data, 'Yearly_equip_failure')

print("Upper limit for Yearly_equip_failure is" , upper_yef)

print("Lower limit for Yearly_equip_failure is" , lower_yef)

data.Yearly_equip_failure = np.where(data.Yearly_equip_failure > upper_yef,
upper_yef,
                      np.where(data.Yearly_equip_failure < lower_yef, lower_yef,
                      data.Yearly_equip_failure))


#re-boxplotting all variables showing outliers

boxplot=sns.boxplot(x='Population',data=df)

plt.show()

boxplot=sns.boxplot(x='Children',data=df)

plt.show()

boxplot=sns.boxplot(x='Income',data=df)

plt.show()

boxplot=sns.boxplot(x='Outage_sec_perweek',data=df)

plt.show()

boxplot=sns.boxplot(x='Email',data=df)

plt.show()

boxplot=sns.boxplot(x='Contacts',data=df)

plt.show()
```

```
boxplot=sns.boxplot(x='Yearly_equip_failure',data=df)
plt.show()
```

The next element of the data preparation process was the data exploration phase. This phase included the univariate and bivariate statistics on the explanatory variables and identifying high cardinality. High cardinality was found by using the .nunique() function. This function provided the count of unique values within each variable. The categorical variables with high cardinality and variables that were deemed unnecessary to the research thesis were removed. High cardinality was viewed as values over five. All unneeded variables were dropped using the .drop() function.

The computation of univariate statistics was applied to the remaining explanatory variables. Distribution plots from the Seaborn library were utilized in visualizing the quantitative variables of “Population”, “Children”, “Age”, “Income”, “Outage_sec_perweek”, “Contacts”, “Yearly_equip_failure”, “Tenure”, and “MonthlyCharge”. The categorical explanatory variables’ univariate statistics were shown via bar plots. The variable’s unique values were grouped and then plotted to allow for the bar plot to be formed. This was completed by using the .groupby() and .size() functions. The Matplotlib library as an inline action was used in providing the visualization of the specified data. The following categorical variables that were plotted included: “Area”, “Marital”, “Gender”, “Contract”, “InternetService”, and “PaymentMethod”.

The bivariate statistics were visualized in a couple of diverse ways. The Seaborn .boxplot() function was applied to the quantitative variables to graphically visualize the distributions as the target variable was categorical. The x-axis value was the target variable of “Churn”. The y-axis values were changed to each of the explanatory variables. The second visualization of bivariate statistics was completed on the categorical variables. These were first passed through a cross-tabular or crosstab table. The results were graphically visualized via a Pandas bar plot to show the distributions.

Below was the annotated code for the data exploration phase.

```
print(f'CaseOrder: {data.CaseOrder.nunique()}')
print(f'Customer_id: {data.Customer_id.nunique()}')
print(f'Interaction: {data.Interaction.nunique()}')
print(f'UID: {data.UID.nunique()}')
print(f'City: {data.City.nunique()}')
print(f'State: {data.State.nunique()}')
print(f'County: {data.County.nunique()}')
print(f'Zip: {data.Zip.nunique()}')
print(f'Lat: {data.Lat.nunique()}')
print(f'Lng: {data.Lng.nunique()}')
print(f'Population: {data.Population.nunique()}')
print(f'Area: {data.Area.nunique()}')
print(f'TimeZone: {data.TimeZone.nunique()}')
print(f'Job: {data.Job.nunique()}')
print(f'Children: {data.Children.nunique()}')
print(f'Age: {data.Age.nunique()}')
print(f'Income: {data.Income.nunique()}')
print(f'Marital: {data.Marital.nunique()}')
print(f'Gender: {data.Gender.nunique()}')
print(f'Churn: {data.Churn.nunique()}')
print(f'Outage_sec_perweek: {data.Outage_sec_perweek.nunique()}')
print(f'Email: {data.Email.nunique()}')
print(f'Contacts: {data.Contacts.nunique()}')
print(f'Yearly_equip_failure: {data.Yearly_equip_failure.nunique()}')
print(f'Techie: {data.Techie.nunique()}')
print(f'Contract: {data.Contract.nunique()}')
print(f'Port_modem: {data.Port_modem.nunique()}')
print(f'Tablet: {data.Tablet.nunique()}')
print(f'InternetService: {data.InternetService.nunique()}')
print(f'Phone: {data.Phone.nunique()}')
print(f'Multiple: {data.Multiple.nunique()}')
print(f'OnlineSecurity: {data.OnlineSecurity.nunique()}')
print(f'OnlineBackup: {data.OnlineBackup.nunique()}')
print(f'DeviceProtection: {data.DeviceProtection.nunique()}')
print(f'TechSupport: {data.TechSupport.nunique()}')
print(f'StreamingTV: {data.StreamingTV.nunique()}')
print(f'StreamingMovies: {data.StreamingMovies.nunique()}')
print(f'PaperlessBilling: {data.PaperlessBilling.nunique()}')
print(f'PaymentMethod: {data.PaymentMethod.nunique()}')
print(f'Tenure: {data.Tenure.nunique()}')
print(f'MonthlyCharge: {data.MonthlyCharge.nunique()}')
print(f'Bandwidth_GB_Year: {data.Bandwidth_GB_Year.nunique()}')
print(f'Timely_Respd: {data.Timely_Respd.nunique()}')
print(f'Timely_Fixes: {data.Timely_Fixes.nunique()}')
print(f'Timely_Replc: {data.Timely_Replc.nunique()}')
print(f'Reliability: {data.Reliability.nunique()}')
print(f'Options: {data.Options.nunique()}')
print(f'Respect_Resp: {data.Respect_Resp.nunique()}')
print(f'Courteous_Exch: {data.Courteous_Exch.nunique()}')
print(f'Evidence_ActListen: {data.Evidence_ActListen.nunique()}')

#dropping categorical variables with high cardinality and unneeded variables
df.drop(['CaseOrder', 'Customer_id', 'Interaction', 'UID', 'City', 'State', 'Email',
'PaperlessBilling',
    'County', 'Zip', 'Lat', 'Lng', 'TimeZone', 'Job','Bandwidth_GB_Year','Techie',
'Multiple', 'Port_modem', 'Tablet',

    'Phone', 'OnlineSecurity', 'OnlineBackup','DeviceProtection', 'TechSupport',

     'StreamingTV', 'StreamingMovies','Timely_Respd', 'Timely_Fixes',
'Timely_Replc',

    'Reliability', 'Options', 'Respect_Resp', 'Courteous_Exch',

    'Evidence_ActListen'], axis=1, inplace=True)


#univariate statistics, via distplots for quantitative explanatory variables

sns.displot(df['Population'], kde=False, color='blue', bins=7)

sns.displot(df['Children'], kde=False, color='red', bins=7)

sns.displot(df['Age'], kde=False, color='green', bins=7)

sns.displot(df['Income'], kde=False, color='gray', bins=7)

sns.displot(df['Outage_sec_perweek'], kde=False, color='orange', bins=7)

sns.displot(df['Contacts'], kde=False, color='black', bins=5)

sns.displot(df['Yearly_equip_failure'], kde=False, color='purple', bins=5)

sns.displot(df['Tenure'], kde=False, color='pink', bins=7)

sns.displot(df['MonthlyCharge'], kde=False, color='brown', bins=7)


# univariate statistics, via bar plots for categorical explanatory variables

groupedArea = df.groupby(by='Area').size()

groupedArea

%matplotlib inline

groupedArea.plot.bar()

groupedMarital = df.groupby(by='Marital').size()

groupedMarital

%matplotlib inline

groupedMarital.plot.bar(color='brown')

groupedGender = df.groupby(by='Gender').size()

groupedGender
```

```
%matplotlib inline
groupedGender.plot.bar(color='green')
groupedContract = df.groupby(by='Contract').size()
groupedContract
%matplotlib inline
groupedContract.plot.bar(color='pink')
groupedInternetService = df.groupby(by='InternetService').size()
groupedInternetService
%matplotlib inline
groupedInternetService.plot.bar(color='red')
groupedPayment = df.groupby(by='PaymentMethod').size()
groupedPayment
%matplotlib inline
groupedPayment.plot.bar(color='orange')

#bivariate statistics visualizations
# 2 categorical variables using crosstab table and bar plot
 ca_crosstab=pd.crosstab(index=df['Churn'], columns=df['Area'])
print(ca_crosstab)am_scatter.set_ylabel('Age')
ca_crosstab.plot.bar(figsize=(7,4), rot=0)
cm_crosstab=pd.crosstab(index=df['Churn'], columns=df['Marital'])
print(cm_crosstab)
cm_crosstab.plot.bar(figsize=(7,4), rot=0)
cg_crosstab=pd.crosstab(index=df['Churn'], columns=df['Gender'])
print(cg_crosstab)
cg_crosstab.plot.bar(figsize=(7,4), rot=0)
cc_crosstab=pd.crosstab(index=df['Churn'], columns=df['Contract'])
print(cc_crosstab)
cc_crosstab.plot.bar(figsize=(7,4), rot=0)
ci_crosstab=pd.crosstab(index=df['Churn'], columns=df['InternetService'])
```

```
print(ci_crosstab)
ci_crosstab.plot.bar(figsize=(7,4), rot=0)
cpm_crosstab=pd.crosstab(index=df['Churn'], columns=df['PaymentMethod'])
print(cpm_crosstab)
cpm_crosstab.plot.bar(figsize=(7,4), rot=0)

# bivariate statistics of categorical/continuous variable via boxplots
sns.boxplot(data=df, x="Churn", y="Population ")
sns.boxplot(data=df, x="Churn ", y="Children ")
sns.boxplot(data=df, x="Churn ", y="Age ")
sns.boxplot(data=df, x="Churn ", y="Income ")
sns.boxplot(data=df, x="Churn ", y="Outage_sec_perweek ")
sns.boxplot(data=df, x="Churn ", y="Contacts ")
sns.boxplot(data=df, x="Churn ", y="Yearly_equip_failure ")
sns.boxplot(data=df, x="Churn ", y="Tenure ")
sns.boxplot(data=df, x="Churn ", y="MonthlyCharge ")
```

The last step of the data preparation phase was the data wrangling phase. During this phase, the categorical variables are re-expressed into numerical variables. The completion of this was done in two ways. The categorical variables which have only two values are converted using ordinal encoding. The ordinal encoding method used the cat.codes function to change each “No” value to “0” and each “Yes” to “1”. In this dataset, only the target variable of “Churn” was changed in this way.

The remaining categorical explanatory variables were nominal with more than two unique values. These were converted using the Pandas get_dummies method. The nominal attributes were passed through this function to create dummy variables of the original attributes. This was done by concatenating the original variable name with each unique value found to create a new dummy variable. It was further expanded by putting a “1” in the column of the dummy variable that was previously referenced by the original variable. After the completion of the get_dummies computation, one of the dummy variables of each grouping was dropped to reduce multicollinearity.

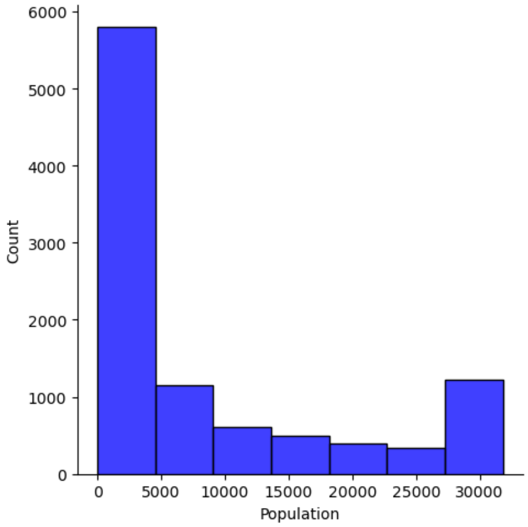
The final step of the data wrangling process was to check for multicollinearity. The variance inflation factor (VIF) method was utilized to complete this. The VIF value over ten has a high likelihood of multicollinearity. In this dataset, there were two variables with a VIF score of greater than ten. The first one was “MonthlyCharge” with a value of 13.51. The second was “Outage_sec_perweek” with a VIF value of 10.05. Both of the variables were dropped from the dataset using the .drop() function.

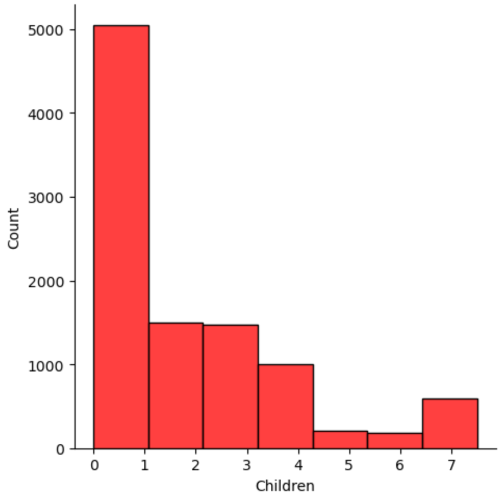
Below was the annotated code for the data wrangling phase.

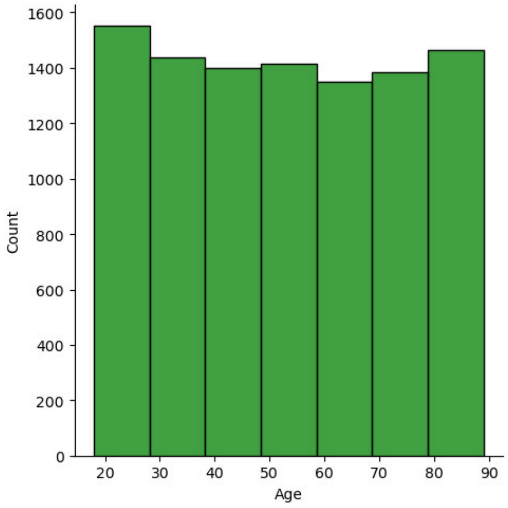
```
#re-expression of categorical variables
#convert ordinal categorical to numerical
data['Churn']=data['Churn'].astype('category')
data['Churn']=data['Churn'].cat.codes

#utilizing get_dummies to convert nominal categorical to numerical
df= pd.get_dummies(df, columns=['Area', 'Marital', 'Gender', 'Contract',
                              'InternetService', 'PaymentMethod'], prefix_sep=" " ,
drop_first=True)

#VIF to check for multicollinearity, if greater than 10,  drop variables.
X=df[['Population', 'Children', 'Age', 'Income',
    'Outage_sec_perweek', 'Contacts', 'Yearly_equip_failure',
    'Tenure', 'MonthlyCharge', 'Area Suburban', 'Area Urban', 'Marital Married',
    'Marital Never Married', 'Marital Separated', 'Marital Widowed',
    'Gender Male', 'Gender Nonbinary', 'Contract One year',
    'Contract Two Year', 'InternetService Fiber Optic',
    'InternetService None', 'PaymentMethod Credit Card (automatic)',
    'PaymentMethod Electronic Check', 'PaymentMethod Mailed Check']]
vif_data = pd.DataFrame()
vif_data["Explanatory Variables"] = X.columns

vif_data["VIF"] = [variance_inflation_factor(X.values, i)
for i in range(len(X.columns))]

vif_data["VIF"]=round(vif_data["VIF"],2)
vif_data=vif_data.sort_values(by="VIF", ascending=False)
print(vif_data)

#dropping variables with high multicollinearity
df.drop(['MonthlyCharge', 'Outage_sec_perweek'], axis=1, inplace=True)
```

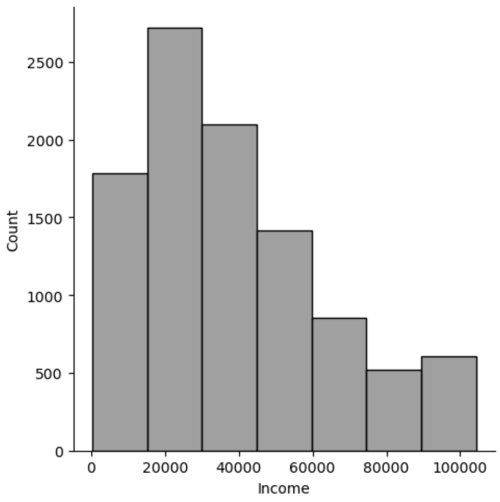
C4. Visualizations of Distributions via Univariate and Bivariate Statistics

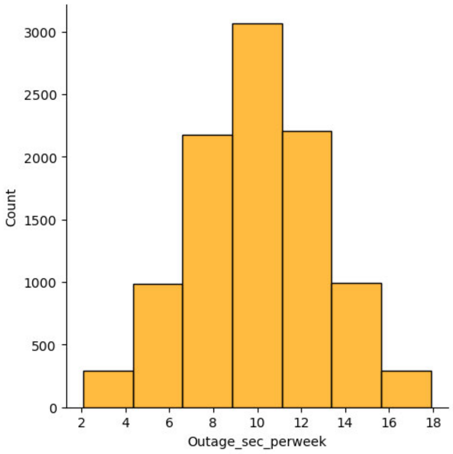
Please see below for the univariate and bivariate analysis visualizations for each of the explanatory variables. The univariate statistics findings were visualized for the quantitative variables, “Population”, “Children”, “Age”, “Income”, “Outage_sec_perweek”, “Contacts”, “Yearly_equip_failure”, “Tenure”, and “MonthlyCharge”. These variables were plotted as distplots, which were distribution plots in histogram form. The categorical variables of “Area”, “Marital”, “Gender”, “Contract”, “InternetService”, and “PaymentMethod” were plotted via bar plots to visualize the distributions.

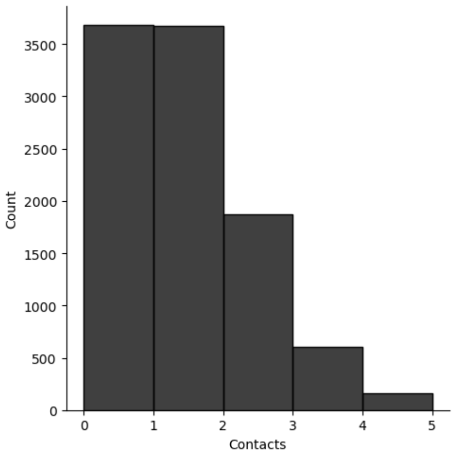

Count
0
1000
2000
3000
4000
5000
6000
0
5000
10000
15000
20000
25000
30000
Population

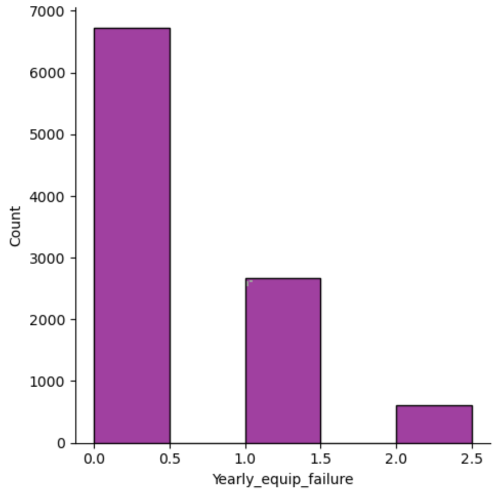

Count
0
1000
2000
3000
4000
5000
0
1
2
3
4
5
6
7
Children

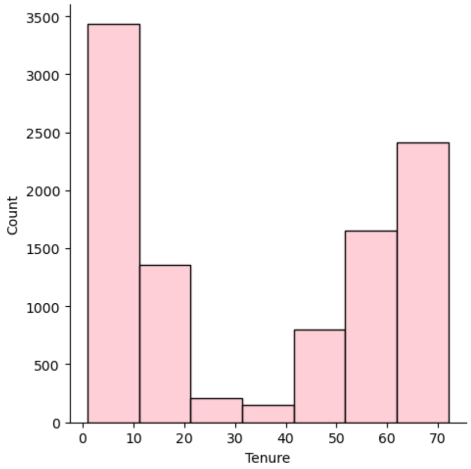

Count
0
200
400
600
800
1000
1200
1400
1600
20
30
40
50
60
70
80
90
Age

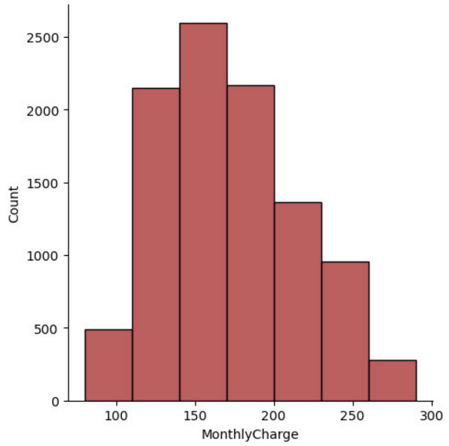

Count
0
500
1000
1500
2000
2500
0
20000
40000
60000
80000
100000
Income

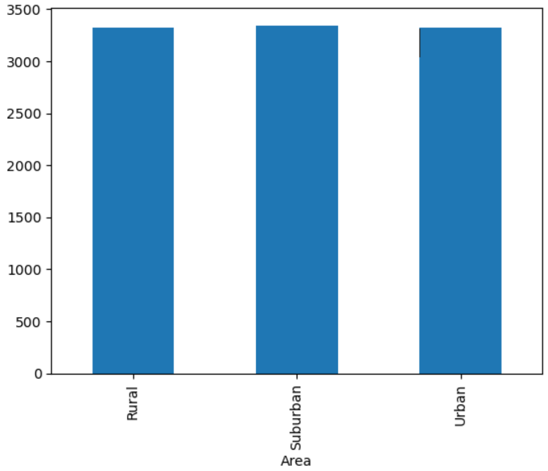

Count
0
500
1000
1500
2000
2500
3000
2
4
6
8
10
12
14
16
18
Outage_sec_perweek

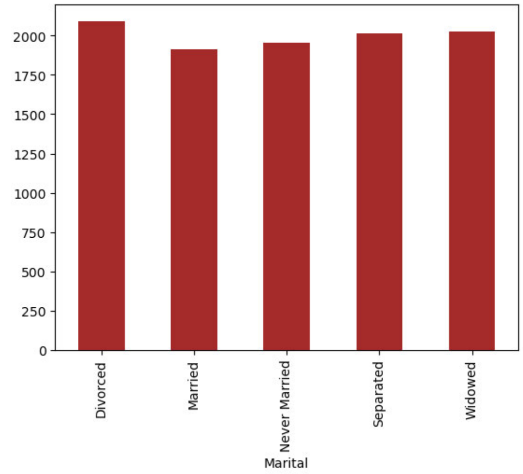

Count
0
500
1000
1500
2000
2500
3000
3500
0
1
2
3
4
5
Contacts

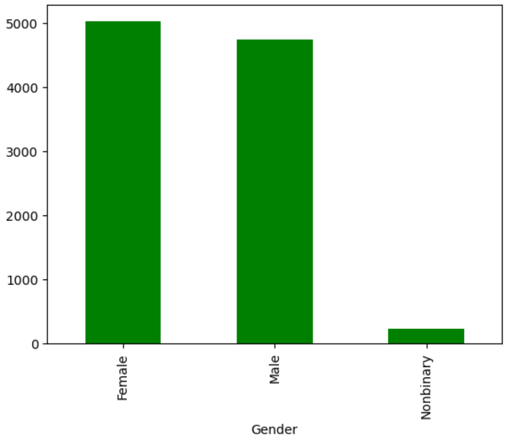

7000
6000
5000
4000
3000
2000
1000
0
Count
0.0
0.5
1.0
1.5
2.0
2.5
Yearly_equip_failure

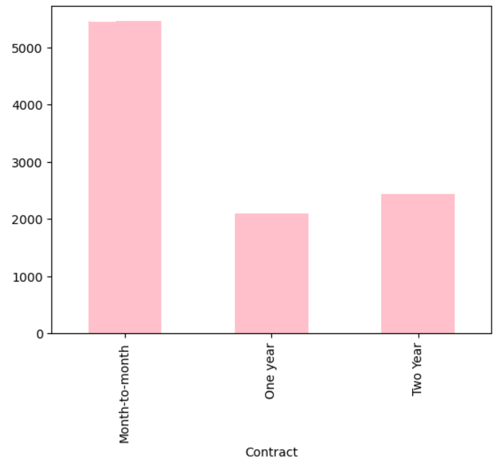

3500
3000
2500
2000
1500
1000
500
0
Count
0
10
20
30
40
50
60
70
Tenure

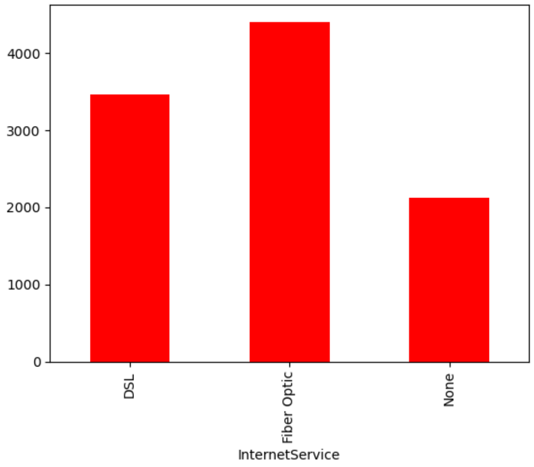

2500
2000
1500
1000
500
0
Count
100
150
200
250
300
MonthlyCharge

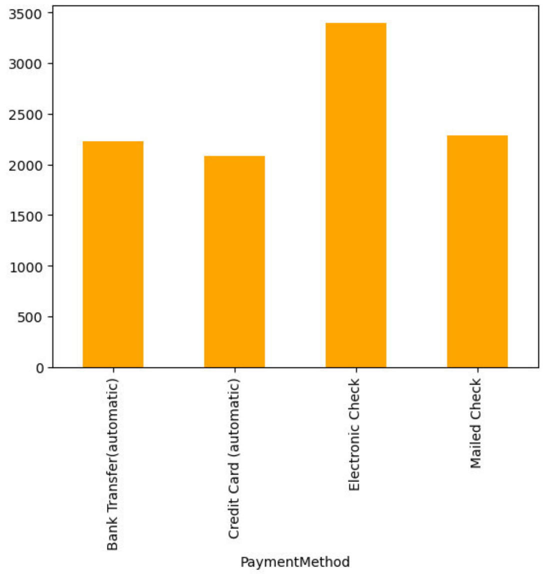

3500
3000
2500
2000
1500
1000
500
0
Rural
Suburban
Urban
Area


2000
1750
1500
1250
1000
750
500
250
0
Divorced
Married
Never Married
Separated
Widowed
Marital

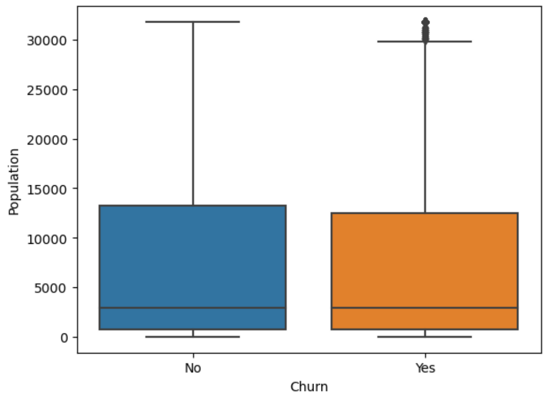

5000
4000
3000
2000
1000
0
Female
Male
Nonbinary
Gender

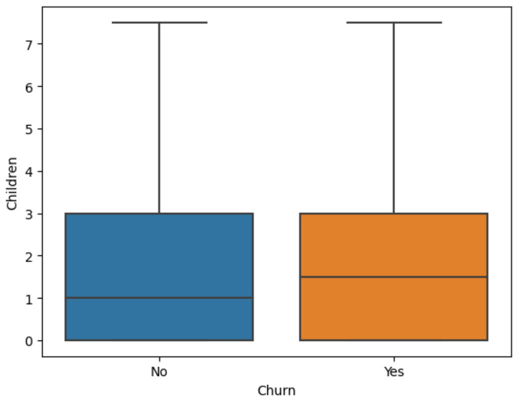


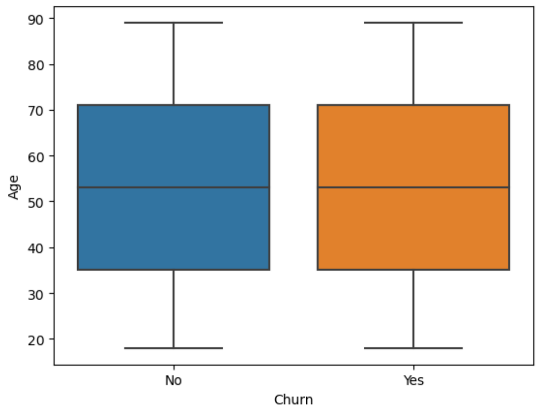


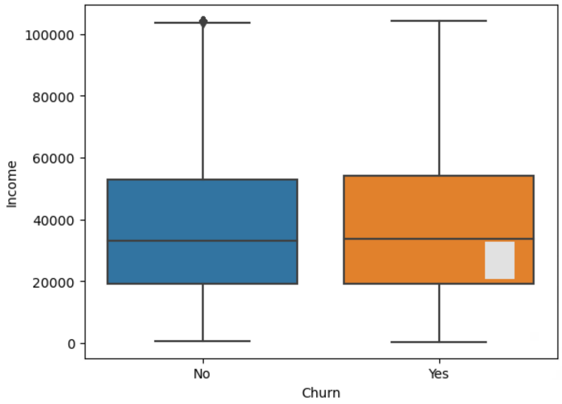


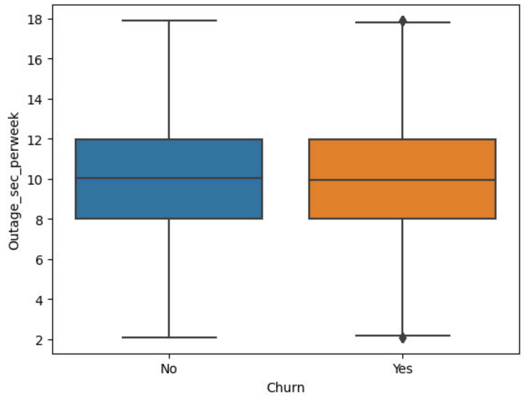
The bivariate statistics were visualized in a couple of ways. The visualizations were based on the type of variables being compared to the target variable of "Churn". The quantitative variables were plotted via boxplots to show the distributions. The remaining categorical variables were computed via a cross-tabular table and further plotted on a bar plot.

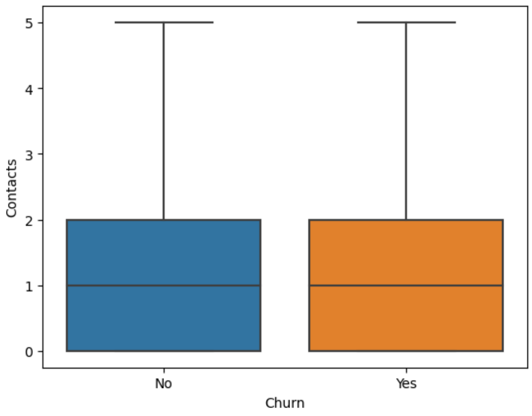

30000
25000
20000
15000
10000
5000
0
Population
No
Yes
Churn

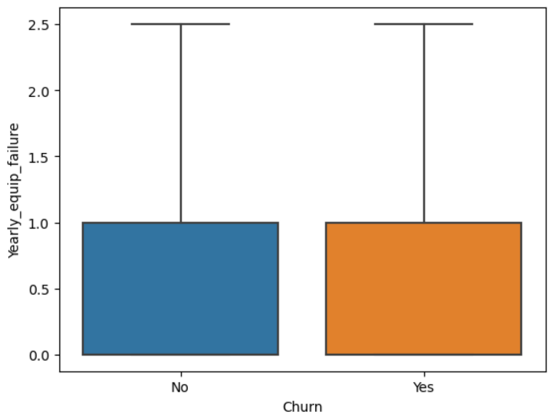

7
6
5
4
3
2
1
0
Children
No
Yes
Churn

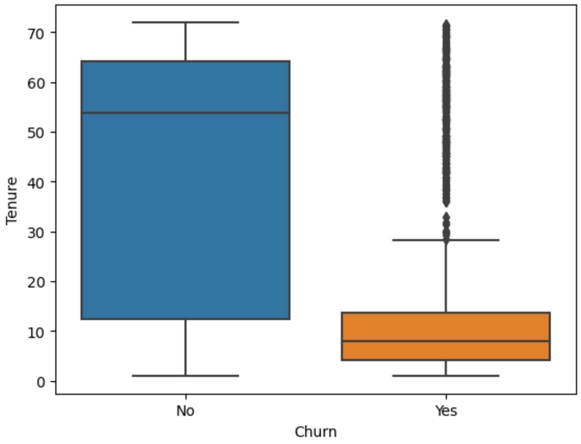

90
80
70
60
50
40
30
20
Age
No
Yes
Churn

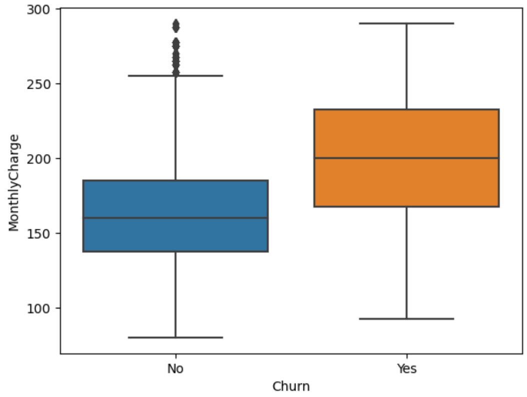

100000
80000
60000
40000
20000
0
Income
No
Yes
Churn

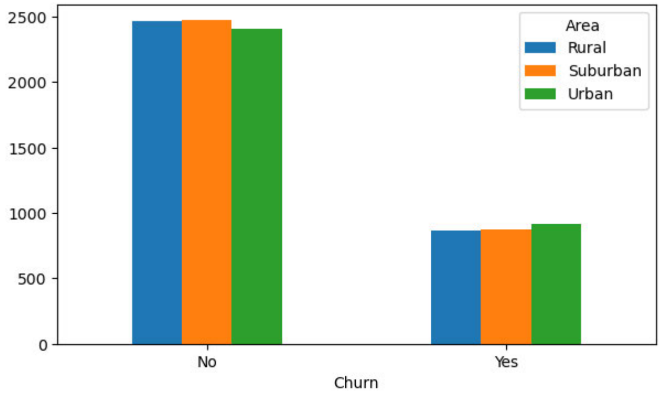

18
16
14
12
10
8
6
4
2
Outage_sec_perweek
No
Yes
Churn

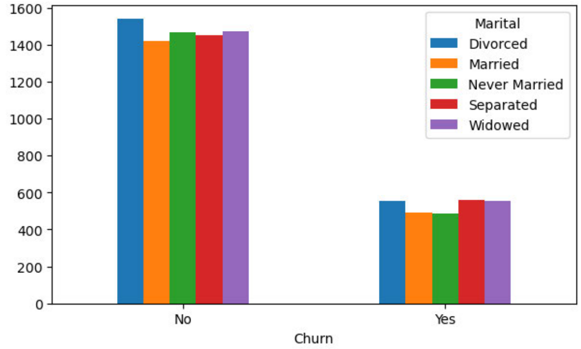

5
4
3
2
1
0
Contacts
No
Yes
Churn

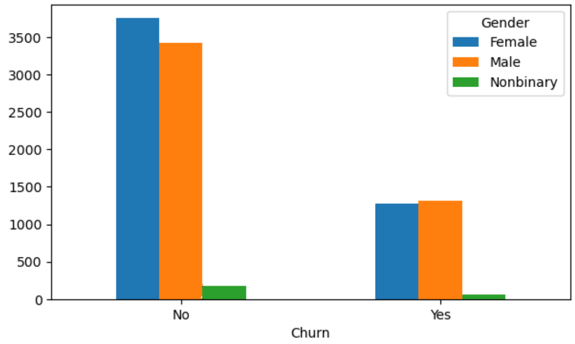

2.5
2.0
1.5
1.0
0.5
0.0
Yearly_equip_failure
No
Yes
Churn

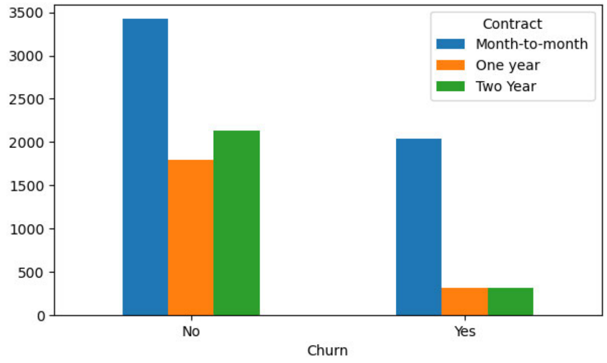

70
60
50
40
30
20
10
0
Tenure
No
Yes
Churn

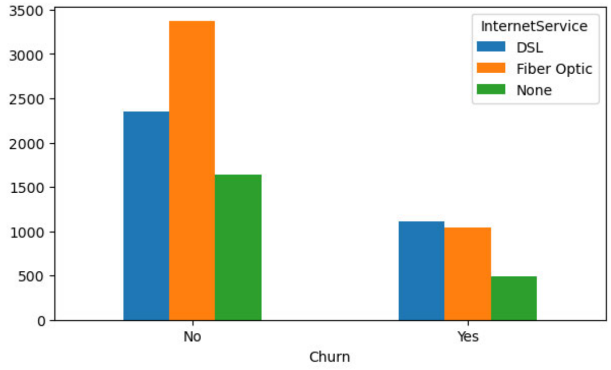

300
250
200
150
100
MonthlyCharge
No
Yes
Churn

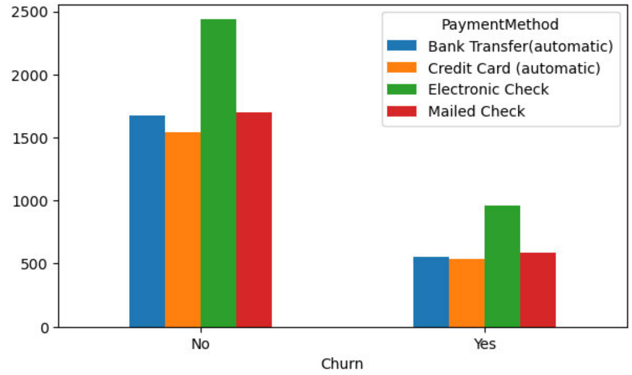

2500
2000
1500
1000
500
0
Area
Rural
Suburban
Urban
No
Yes
Churn


1600
1400
1200
1000
800
600
400
200
0
Marital
Divorced
Married
Never Married
Separated
Widowed
No
Yes
Churn


3500
3000
2500
2000
1500
1000
500
0
Gender
Female
Male
Nonbinary
No
Yes
Churn


3500
3000
2500
2000
1500
1000
500
0
Contract
Month-to-month
One year
Two Year
No
Yes
Churn


3500
3000
2500
2000
1500
1000
500
0
InternetService
DSL
Fiber Optic
None
No
Yes
Churn



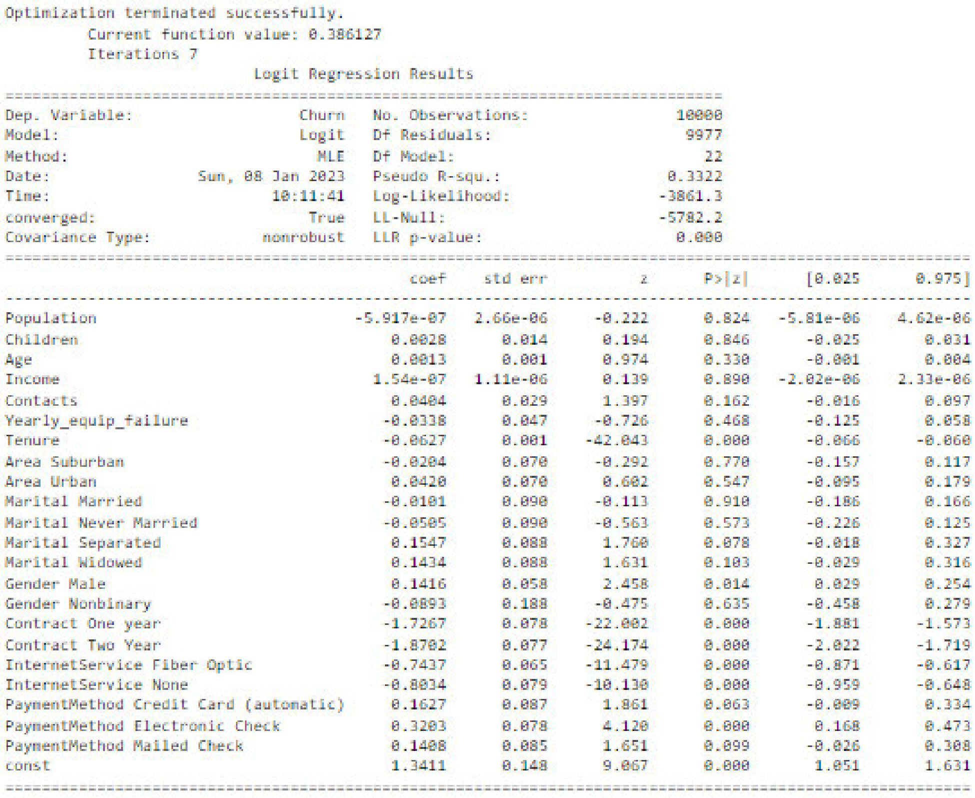
C5. Prepared Dataset

Please see the attached CSV file called AFCode208Task2_clean.csv to view the results of the prepared dataset.

Part IV. Model Comparison and Analysis

D1. Initial Logistic Regression Model

The target variable and re-expressed explanatory variables were analyzed in a logistic regression model via logit. A regression summary was provided by this model based on the target variable, “Churn”. After the data preparation phases, the dataset was left with one categorical target variable and twenty- two explanatory variables. The independent variables included seven continuous and fifteen categorical re-expressed attributes.



```
Optimization terminated successfully.
         Current function value: 0.386127
         Iterations 7
                           Logit Regression Results
==============================================================================
Dep. Variable:                  Churn   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9977
Method:                           MLE   Df Model:                           22
Date:                Sun, 08 Jan 2023   Pseudo R-squ.:                  0.3322
Time:                        10:11:41   Log-Likelihood:                -3861.3
converged:                       True   LL-Null:                       -5782.2
Covariance Type:            nonrobust   LLR p-value:                     0.000
================================================================================================
                                           coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------------------------
Population                           -5.917e-07   2.66e-06     -0.222      0.824   -5.81e-06    4.62e-06
Children                                 0.0028      0.014      0.194      0.846      -0.025       0.031
Age                                      0.0013      0.001      0.974      0.330      -0.001       0.004
Income                                 1.54e-07   1.11e-06      0.139      0.890   -2.02e-06    2.33e-06
Contacts                                 0.0404      0.029      1.397      0.162      -0.016       0.097
Yearly_equip_failure                    -0.0338      0.047     -0.726      0.468      -0.125       0.058
Tenure                                  -0.0627      0.001    -42.043      0.000      -0.066      -0.060
Area Suburban                           -0.0204      0.070     -0.292      0.770      -0.157       0.117
Area Urban                               0.0420      0.070      0.602      0.547      -0.095       0.179
Marital Married                         -0.0101      0.090     -0.113      0.910      -0.186       0.166
Marital Never Married                   -0.0505      0.090     -0.563      0.573      -0.226       0.125
Marital Separated                        0.1547      0.088      1.760      0.078      -0.018       0.327
Marital Widowed                          0.1434      0.088      1.631      0.103      -0.029       0.316
Gender Male                              0.1416      0.058      2.458      0.014       0.029       0.254
Gender Nonbinary                        -0.0893      0.188     -0.475      0.635      -0.458       0.279
Contract One year                       -1.7267      0.078    -22.002      0.000      -1.881      -1.573
Contract Two Year                       -1.8702      0.077    -24.174      0.000      -2.022      -1.719
InternetService Fiber Optic             -0.7437      0.065    -11.479      0.000      -0.871      -0.617
InternetService None                    -0.8034      0.079    -10.130      0.000      -0.959      -0.648
PaymentMethod Credit Card (automatic)    0.1627      0.087      1.861      0.063      -0.009       0.334
PaymentMethod Electronic Check           0.3203      0.078      4.120      0.000       0.168       0.473
PaymentMethod Mailed Check               0.1408      0.085      1.651      0.099      -0.026       0.308
const                                    1.3411      0.148      9.067      0.000       1.051       1.631
================================================================================================
```

D2. Justification of Variable Selection and Evaluation Metric

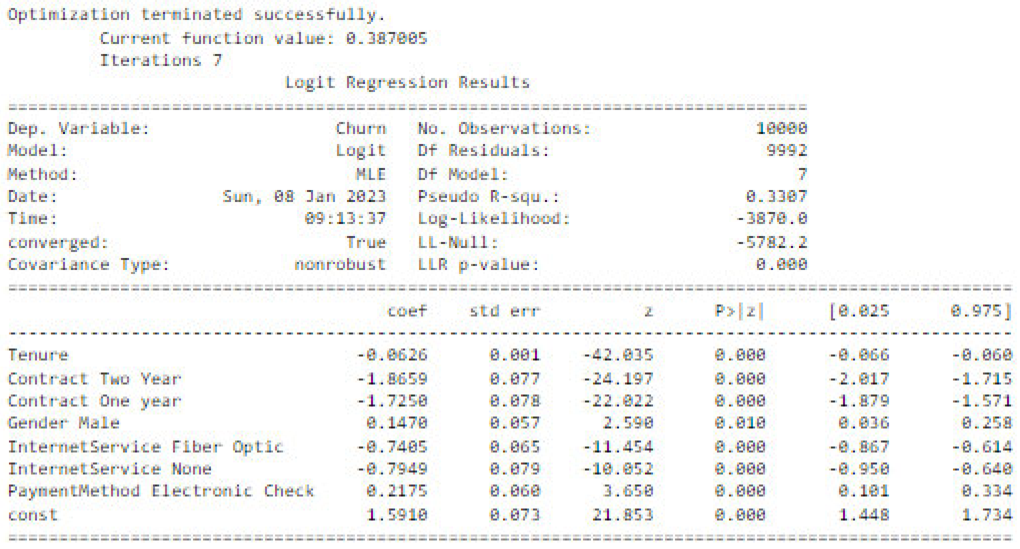
The selection of the explanatory variables to eliminate for the reduced logistic regression model was made by implementing a wrapper method on the initial model. The wrapper method chosen was the Backward Stepwise Elimination method. This variable selection method utilized the initial regression model to review the p-values of the explanatory variables. In the initial model, the p-value was found in the P> |z| column. The Backward Stepwise Elimination method was justified as it kept only the features that had a p-value that was deemed statistically significant, which was noted as a lesser than 0.05 value. The continuous variable which met this criterion was “Tenure”. The categorical variables included “Gender Male”, “Contract One year”, “Contract Two Year”, “InternetService Fiber Optic”, “InternetService None”, “PaymentMethod

Electronic Check", and the constant. All of these features excluding "Gender Male" had a p-value of 0.0. The variable "Gender Male" had a p-value of 0.014.

The initial model and the reduced model showed an evaluation metric, Pseudo R-squared value of around 33%. There was a slight deviation between them with the initial model Pseudo R-squared value of 0.3322 and the reduced model with a 0.3307 value. This value signified the initial model as a slightly better fit. The second evaluation metric utilized was the LLR p-value. This metric provided a conclusion that the regression model overall was useful compared to a model with no explanatory variables. The metric had to be less than 0.05 to be deemed useful. Both the initial and the reduced models had an LLR p-value of 0.00, which met this criterion.

D3. Reduced Logistic Regression Model

The reduced logistic regression model was decreased by the application of the variable selection method, Backward Stepwise Elimination. The explanatory variables remaining for the reduced regression model were "Tenure", "Contract One year", "Contract Two Year", "Gender Male". "InternetService Fiber Optic", "InternetService None", and "PaymentMethod Electronic Check". The only continuous variable incorporated in the reduced model was "Tenure"; the six other variables were categorical.



```
Optimization terminated successfully.
         Current function value: 0.387005
         Iterations 7
                           Logit Regression Results                           
==============================================================================
Dep. Variable:                  Churn   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9992
Method:                           MLE   Df Model:                            7
Date:                Sun, 08 Jan 2023   Pseudo R-squ.:                  0.3307
Time:                        09:13:37   Log-Likelihood:                -3870.0
converged:                       True   LL-Null:                       -5782.2
Covariance Type:            nonrobust   LLR p-value:                     0.000
=================================================================================================
                                      coef    std err          z      P>|z|     [0.025     0.975]
-------------------------------------------------------------------------------------------------
Tenure                             -0.0626      0.001    -42.035      0.000     -0.066     -0.060
Contract Two Year                  -1.8659      0.077    -24.197      0.000     -2.017     -1.715
Contract One year                  -1.7250      0.078    -22.022      0.000     -1.879     -1.571
Gender Male                         0.1470      0.057      2.590      0.010      0.036      0.258
InternetService Fiber Optic        -0.7405      0.065    -11.454      0.000     -0.867     -0.614
InternetService None               -0.7949      0.079    -10.052      0.000     -0.950     -0.640
PaymentMethod Electronic Check      0.2175      0.060      3.650      0.000      0.101      0.334
const                               1.5910      0.073     21.853      0.000      1.448      1.734
=================================================================================================
```

Part IV. Data Analysis: Outcome

E1. Discussion of Data Analysis Process

The data analysis procedure for comparison of the initial and reduced logistic regression model included the following elements: the variable selection technique and the model evaluation metrics. The variable selection technique was performed via a wrapper method. The wrapper method was called Backward Stepwise Elimination. It reduced the initial model by removing variables that met the pre-specified stop value of p-value less than 0.05. The initial model included the following variables: “Population”, “Children”, “Age”, “Income”, “Contacts”, “Yearly_equip_failure”, “Tenure”, “Area Suburban”, “Area Urban”, “Marital Married”, “Married Never Married”, “Marital Separated”, “Marital Widowed”, “Gender Male”, “Gender Nonbinary”, “Contract One year”, “Contract Two Year”, “InternetService Fiber Optic”, “InternetService None”, “PaymentMethod Credit Card (automatic)”, “PaymentMethod Electronic Check”, “PaymentMethod Mailed Check”, and “const”. The variable selection method was completed, and the following variables met the stop rule specified: “Tenure”, “Gender Male”, “Contract One year”, “Contract Two Year”, “InternetService Fiber Optic”, “InternetService None”, “PaymentMethod Electronic Check”, and “const”. Those specified were the variables included in the reduction model.

The model evaluation metrics shown in the initial and reduced regression models were Log-Likelihood, Pseudo R squared, and LLR-p-value. The Log-Likelihood was a value of the regression model to measure the goodness of fit (Broeck, Intermediate Regression with statsmodels in Python, n.d.). The initial model’s Log-likelihood has a higher value compared to the reduced model so it should be utilized for further analysis. The Pseudo R-squared value also measured the fit, but the measurement was the fit of the model compared to another model. The higher the value the better the fit. The same determination could be made that the initial model was a better fit in comparison as it had a higher Pseudo R-squared value by 0.0015. The last evaluation metric was the LLR-p-value. The LLR-p-value was the p-value for the complete model. This was deemed useful as long as the value was under the threshold of the statistically significant level. The significance level value was a p-value of under 0.05. Both of the regression models met this criterion. The above metrics helped identify which model was better useful for predictive modeling.

The confusion matrix was also a metric utilized in analyzing the regression models. The confusion matrix provided an array of values noting the true positives, false positives, true negatives, and false negatives of the predictive models. The initial and reduced regression models had similar confusion matrices. The initial model’s confusion matrix resulted in the following statistics:

True positive predictions: 1949

True negative predictions: 487

False positive predictions: 312

False negative predictions: 252

The reduced model's confusion matrix resulted in the following:

True positive predictions: 1948

True negative predictions: 492

False positive predictions: 307

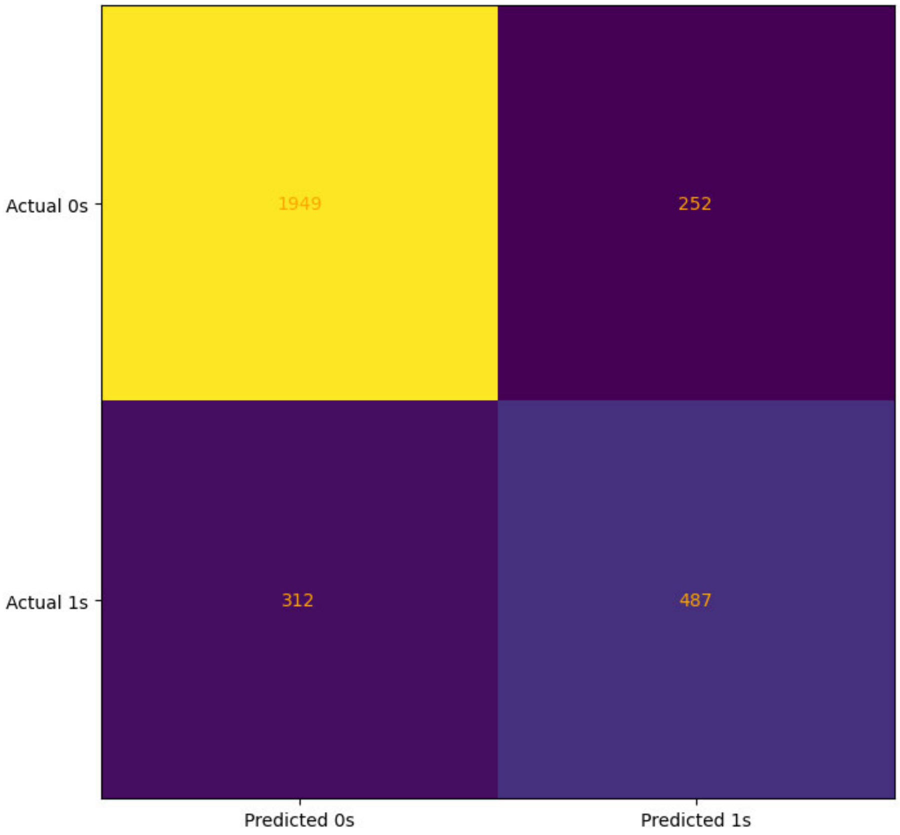
False negative predictions: 253

Overall, the reduced model's confusion matrix was a better representation as it had a higher value of correctly predicted values.

E2. Data Analysis Output Results

The results of the data analysis calculations as well as the confusion matrix were shown below.

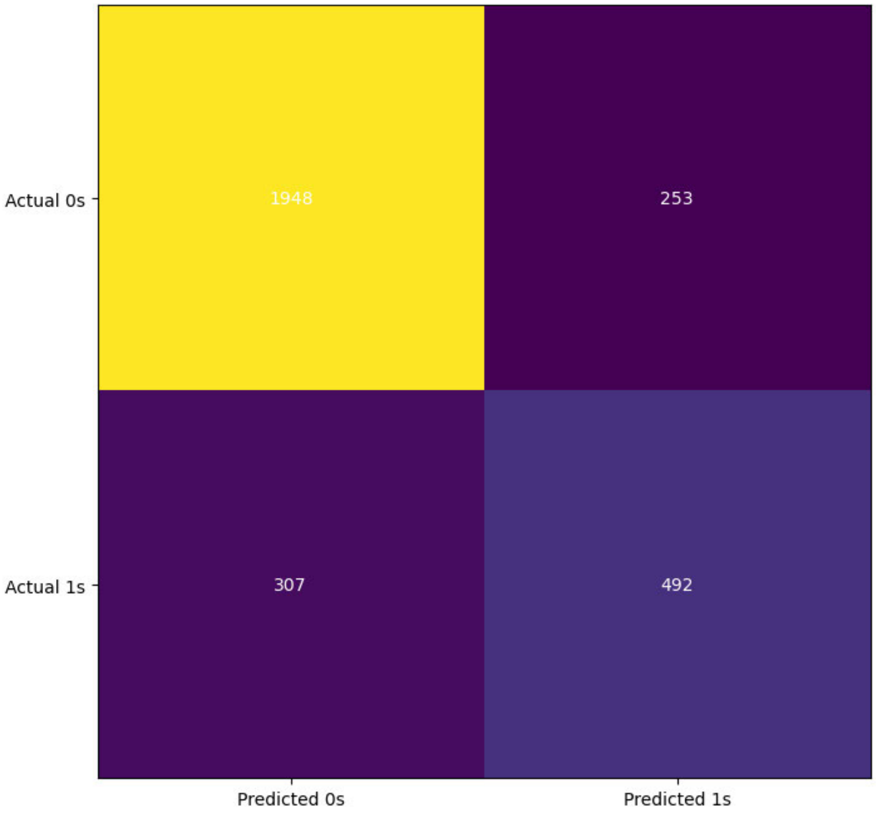
```
0.8227142857142857

array([[1949,  252],
[ 312,  487]], dtype=int64)
```



```
              precision    recall  f1-score   support

           0       0.86      0.89      0.87      2201
           1       0.66      0.61      0.63       799

    accuracy                           0.81      3000
   macro avg       0.76      0.75      0.75      3000
weighted avg       0.81      0.81      0.81      3000
```

```
0.8133333333333334
```

```
array([[1948,  253],
       [ 307,  492]], dtype=int64)
```



```
              precision    recall  f1-score   support

           0       0.86      0.89      0.87      2201
           1       0.66      0.62      0.64       799

    accuracy                           0.81      3000
   macro avg       0.76      0.75      0.76      3000
weighted avg       0.81      0.81      0.81      3000
```

E3. Data Analysis Code

```
# logistic regression model
df['const']=1
y= df['Churn']
X= df[['Population', 'Children', 'Age', 'Income',
    'Contacts', 'Yearly_equip_failure', 'Tenure',
    'Area Suburban', 'Area Urban', 'Marital Married',
    'Marital Never Married', 'Marital Separated', 'Marital Widowed',
    'Gender Male', 'Gender Nonbinary', 'Contract One year',
    'Contract Two Year', 'InternetService Fiber Optic',
    'InternetService None', 'PaymentMethod Credit Card (automatic)',
    'PaymentMethod Electronic Check', 'PaymentMethod Mailed Check','const']]
log_model = sm.Logit(y,X)
results = log_model.fit()
print(results.summary())

y= df['Churn']
X= df[['Population', 'Children', 'Age', 'Income',
    'Contacts', 'Yearly_equip_failure', 'Tenure',
    'Area Suburban', 'Area Urban', 'Marital Married',
    'Marital Never Married', 'Marital Separated', 'Marital Widowed',
    'Gender Male', 'Gender Nonbinary', 'Contract One year',
    'Contract Two Year', 'InternetService Fiber Optic',
    'InternetService None', 'PaymentMethod Credit Card (automatic)',
    'PaymentMethod Electronic Check', 'PaymentMethod Mailed Check','const']]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=0)
scaler = RobustScaler()
X_train = scaler.fit_transform(X_train)
log_r= LogisticRegression(random_state=0).fit(X_train, y_train)
y_pred = log_r.predict(X_test)
log_r.score(X_train, y_train)
confusion_matrix(y_test, y_pred)
matrix_i = confusion_matrix(y_test, , y_pred)
fig, ax = plt.subplots(figsize=(8, 8))
ax.imshow(matrix_i)
ax.grid(False)
ax.xaxis.set(ticks=(0, 1), ticklabels=('Predicted 0s', 'Predicted 1s'))
ax.yaxis.set(ticks=(0, 1), ticklabels=('Actual 0s', 'Actual 1s'))
ax.set_ylim(1.5, -0.5)
for i in range(2):
   for j in range(2):
      ax.text(j, i, matrix_i[i, j], ha='center', va='center', color=orange)
plt.show()
print(classification_report(y_test, , y_pred))

#reduced model via backward stepwise elimination with p-values less than 0.05
df['const']=1
y= df['Churn']
X= df[['Tenure', 'Contract Two Year', 'Contract One year', 'Gender Male',
     'InternetService Fiber Optic', 'InternetService None',
     'PaymentMethod Electronic Check', 'const']]
log_model_red = sm.Logit(y,X)
redu_results = log_model_red.fit()
print(redu_results.summary())
y= df['Churn']
X= df[['Tenure', 'Contract Two Year', 'Contract One year', 'Gender Male',
     'InternetService Fiber Optic', 'InternetService None',
     'PaymentMethod Electronic Check', 'const']]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=0)
scaler = RobustScaler()
X_train = scaler.fit_transform(X_train)
X_test= scaler.transform(X_test)
logr= LogisticRegression(random_state=0).fit(X_test, y_test)
logr.score(X_test, y_test)
y_predr= logr.predict(X_test)
confusion_matrix(y_test, y_predr)
matrix = confusion_matrix(y_test, y_predr)
fig, ax = plt.subplots(figsize=(8, 8))
```

```
ax.imshow(matrix)
ax.grid(False)
ax.xaxis.set(ticks=(0, 1), ticklabels=('Predicted 0s', 'Predicted 1s'))
ax.yaxis.set(ticks=(0, 1), ticklabels=('Actual 0s', 'Actual 1s'))
ax.set_ylim(1.5, -0.5)
for i in range(2):
    for j in range(2):
        ax.text(j, i, matrix[i, j], ha='center', va='center', color='white')
plt.show()
print(classification_report(y_test, y_predr))
```

Part V. Data Summary and Implications

F1. Summarization of Findings

The data analysis of the logistic regression model was summarized by the following elements.

1. A regression equation of the reduced model.
2. An interpretation of coefficients deemed statistically significant
3. The practical and statistical significance of the regression model
4. Data Analysis limitations

The regression equation for the reduced model was set up based on the logistic regression equation (Vestuto, n.d.).

$$ln\left(\frac{\hat{p}}{1-\hat{p}}\right) = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + b_4X_4 + b_5X_5 \ldots. + b_pX_p$$

The reduced regression model formed the following regression equation.

$ln\left(\frac{\hat{p}}{1-\hat{p}}\right)$ = 1.591 - 0.06(Tenure) – 1.8659(Contract Two Year) – 1.725(Contract One year) + 0.147 (Male Gender) – 0.7405 (InternetService Fiber Optic) – 0.7949 (InternetService None) + 0.2175(PaymentMethod Electronic Check)

The regression equation utilized the intercept as b_0. In this logistic regression model, the variable "const" was the intercept. Afterward, the regression coefficient estimates for every explanatory variable in the reduced model were added to the intercept to make the logistic regression equation.

The coefficients within the logistic regression model associated with the predictor variable provided the expected change in log odds of having the outcome per unit change in the independent variable (Broeck, Intermediate Regression with statsmodels in Python, n.d.). The "Tenure" variable had a coefficient of -0.0626, which meant for every one unit increase in "Tenure", there was a reduction to log odds of "Churn" by 0.0626. The same rules were followed for the remaining variables. The "Contract Two Year" variable had a coefficient of -1.8659. This was a reduction to log odds of "Churn" by 1.8659 with every increase in one unit of "Contract Two Year". The variable "Contract One year" had a coefficient of -1.7250, which was a reduction to log odds of "Churn" by 1.7250 with one unit increase in "Contract One year". The "Gender Male" variable had a coefficient of 0.1470. This meant for each one unit increase in "Gender Male", there was a change in the log odds of "Churn" by 0.1470. The "InternetService Fiber Optic" variable had a coefficient of -0.7405, which meant that for every unit increase in "InternetService Fiber Optic" there was a reduction in the log odds of "Churn" by 0.7405. The coefficient for the variable "InternetService None" was -0.7949. This was interpreted as a reduction in the log odds of "Churn" by 0.7949 for every unit increase of "InternetService None". The

"PaymentMethod Electronic Check" variable had a coefficient of 0.2175, which meant with every unit increase of the independent variable there was a change in log odds of "Churn" by 0.2175. The intercept was the expected value's mean for the target variable when all explanatory variables are equal to zero (Vestuto, n.d.). With that explained, for every unit increase of the "const" there was an associated 1.591 change of log odds for "Churn".

The initial model was statistically significant. The model was significant because the z-statistic and LLR p-value found in the regression summary provided significant values. The LLR p-value met the threshold for significance. This was deemed as a value lesser than or equal to the value of 0.05. In the initial model, the LLR p-value was 0.0. The z-statistics of the initial model provided a better statistical fit due to the strength of the predictors. The z column in the regression summary noted the z-statistics. When the z-statistic value was further from zero, the stronger its role was as a predictor variable. This was shown in the tenure, contract, and internet service variables. They had a value over ten away from zero.

The model had practical significance. The model results on the telecommunication dataset have real-world importance to the stakeholders. As regression assumptions must be verified to provide significance, the regression model applied to the dataset was all valid and confirmed. The coefficient estimations within model provided a sense of what variables would change the log odds of "Churn". The confusion matrix provided a good practical use in predicting the actual log odds outcomes. It had an 82% accuracy rate on predicting if the independent variables would accurately predict the chance of "Churn". All of this information provided the stakeholders with insight into which attributes should be further investigated in terms of retention (Expert Panel, Forbes Agency Council, 2019).

As with any data analysis, there were various limitations. Logistic Regression was limited to extreme outliers and multicollinearity. The presence of these features could cause the results of the regression model to be unreliable or skewed. The data preparation steps would notate some limitations as well. In the data wrangling stage, the re-expression of variables was limited to the encoding methods utilized. The type of encoding applied would provide inaccurate results for categorical variables, which would make the analysis distorted. Another limitation would be the treatment of outliers. Trimming outliers would provide different results compared to capping the outliers as done in the regression model above. Trimming removed outliers from the dataset. Lastly, the model reduction method limitations were related to the multicollinearity and p-values of the independent variables. Multicollinearity was assessed via the VIF method. This method was sensitive to the variables being calculated. The increase in the number of variables or the inclusion of variables highly related to another variable could increase the multicollinearity and the VIF value. All of the before mentioned points were examples of the numerous limitations of data analysis.

F2. Recommendation of Action

The result of the research question deemed that some variables contributed to the probability of "Churn". The variables that were noted for higher impact based on the z-statistics were "Tenure", "Contract One year", and "Contract Two Year". These three variables noted a stronger predictor fit as they were further away from zero in terms of their z value. The variables notated a reduction of log odds of “Churn”. Other variables found were “Gender Male”, “InternetService Fiber Optic”, “InternetService None”, and “PaymentMethod Electronic Check”. Only the variables “Gender Male” and “PaymentMethod Electronic Check” showed an increase in the log odds of “Churn” with each unit increase in the independent variable.

The recommended course of action based on the results of the research analysis was the suggestion for the telecommunication company to put forth efforts into investigating the reason for the increase in “Churn” found within males and clients that paid via electronic check. The investigation can provide further insight into what retention methods will need to be brainstormed to reduce the likelihood of “Churn”.

Part VI. Demonstration

G. Panopto Video

Please see attached Panopto video link. Link Found here:

[redacted]

This was a video providing a summary of the Python code written in Jupyter Notebook. This code was used for the data cleaning process, univariate and bivariate statistics, and regression modeling with coefficients. The recording demonstrated the code’s warning and error-free functionality. It also summarized an overview of the programming environment, Jupyter Notebook.

H. Third-Party Web Sources

There were no web sources used to acquire data nor segments of third-party code to support the application. All code and segments are original work.

I. References

Bobbitt, Z. (2020, October 13). *The 6 Assumptions of Logistic Regression*. Retrieved from Statology: https://www.statology.org/assumptions-of-logistic-regression/

Broeck, M. V. (n.d.). *Intermediate Regression with statsmodels in Python*. Retrieved from Datacamp: https://app.datacamp.com/learn/courses/intermediate-regression-with-statsmodels-in-python

Broeck, M. V. (n.d.). *Introduction to Regression with statsmodels in Python*. Retrieved from Datacamp: https://app.datacamp.com/learn/courses/introduction-to-regression-with-statsmodels-in-python

Expert Panel, Forbes Agency Council. (2019, December 20). *How To Increase Client Retention: 10 Effective Strategies*. Retrieved from Forbes: https://www.forbes.com/sites/forbesagencycouncil/2019/12/20/how-to-increase-client-retention-10-effective-strategies/?sh=3f663798390b

IBM. (n.d.). *What is logistic regression?* Retrieved from IBM: https://www.ibm.com/topics/logistic-regression

Statistic Solutions. (n.d.). *Assumptions of Logistic Regression*. Retrieved from https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/assumptions-of-logistic-regression/

Vestuto, J. (n.d.). *Introduction to Linear Modeling in Python*. Retrieved from Datacamp: https://app.datacamp.com/learn/courses/introduction-to-linear-modeling-in-python