WGU D208 PA 2

Andrew Shrestha

Part I: Research Question

A1. Question: Are we able to accurately predict which customers are at risk of Churn? Can we identify the significant variables that predict the churn outcome?

A2. Objectives and Goals: There are 2 very important reasons why a company would benefit from tracking its churn rate. First, the loss of customers is equivalent to loss of revenue, and due to this being a driving factor in a company's future growth and development, becomes dire to prevent. Secondly, it is generally more expensive to gain new customers than it is to maintain existing ones, therefore it is in the company's and shareholder's best interest to allocate resources to prevent customer churn **(Zendesk 2021).**

With the knowledge of the significant variables identified and our prediction of which customers will churn, companies will have a strong sense of where to allocate their resources to ensure that customer churn is reduced.

Part II: Method Justification

B1. Assumptions of Logistic Regression Model

i. The Dependent variable is categorical and binary (In this case: "Yes"/"No" Outcome)
ii. All Observations considered are independent
iii. Independent variables are linearly related to the log odds
iv. No Multicollinearity
v. No Outliers
vi. Requires large sample
vii. Only Significant variables are included

B2. Benefits of Chosen Tools

The tools being utilized in this analysis are going to be primarily Jupyter notebooks and Python Programming Language.

The benefit of using Jupyter notebooks is that it is able to clearly show all codes and their related outputs, and has the option of producing results that are segmented so that it is easy for users to understand what is happening at each and every step. In addition to this, Jupyter notebooks can also be used for various programming languages besides python, therefore having familiarity with the majority of users already.

The benefit of using Python is that it gives us access to various useful libraries that include statistics, visualizations, and regression calculations. One primary benefit that is significant to Python is that it is a general-purpose language that is one of the most popular, and therefore boosts familiarity amount data analysts. This in turn makes it easier to share our findings between users. Finally, when compared to R, it seems that Python pandas is also noticeably faster in processing larger csv files (**Enoch Kan 2018).**

B3. Justification for Logistic Regression for the Analysis

Logistic Regression is appropriate for this Analysis due to the fact that the dependent variables fall under a classification area. Classification in this case, is an area of machine learning that predicts which class or category a specific entity belongs to depending on its features. Since our dependent variable is "Churn" with the categorical binary outcomes of either "Yes" or "No", and our features are the explanatory variables we deem statistically significant in our Churn dataset, Logistic Regression as a classification technique is perfect for the analysis **(Mirko Stojiljkovic 2021).**

Part III: Data Preparation

C1. Data Preparation Goals and Manipulation Used

i. Obtaining the churn data set into jupyter notebooks via read_csv code
ii. Address any repetitions, irrelevant, duplicated, inconsistent, or unnecessary data
iii. Impute missing data values with statistically significant calculations to keep data accuracy intact
iv. Identify outliers and remove them if they are more than 3 standard deviations above or below the mean
v. Utilize the logistic regression function to predict our churn outcome with respect to the explanatory variables.

The goal for the preparation stage of the analysis is data cleaning to ensure that we are dealing with accurate and correct data, free from any errors that may influence our prediction outcome via ii.

The goal for the manipulation stage of the analysis is to first ensure that we have a complete and appropriate data set, by means of imputation and disregard for missing values and outliers respectively as shown in step iii.

These preparation and manipulation stages will set us up for success when we then run our logistic regression analysis to predict the effects that explanatory variables will have on our dependent variables "Churn"

C2. Summary Statistics

This dataset consists of 10,000 observations with a total of 50 variables. The type of data present are a mix of continuous numerical variables, categorical variables, and categorical binary variables consisting of variables in the realm of “Yes”/”No” or “Male”/”Female” outputs such as our dependent variable of “Churn”.

Out of the 10,000 observations, we decided to remove a couple variables that were deemed no relation to our target outcome mvariable “Churn”. Thus, CaseOrder, Customer_id, Interaction, UID, City, State, Country,Zip,Lat,Lng,Population, Area, TimeZone, Marital, and PaymentMethod were all removed, leaving only 18 variables left.

In order to ensure that our regression analysis could be completed, we also changed the categorical binary outputs on select variables from yes/no to numeric 1/0.

Finally, when running the describe function for our new cleaned data set, we see that average children = 2, Age = 53, Income = $39806, Outage_Sec_perweek = 10, Email = 12, Contacts = 0.99, Yearly_equip_failure = 0.39, and tenure =34.52.

C3. Steps used to Prepare Data

i. Read the Churn_Clean dataset CSV file via python programming language and create a data frame based off the churn data
ii. View the dataset columns and rows and understand the relationships between the independent variables and the dependent variable “Churn”. Identify which variables are necessary to keep and remove the variables that have no relation to the Churn outcome
iii. Rename survey questions into more appropriate labels instead of just numeric labels, thus making it easier to keep track of and understand
iv. Get initial statistics of the dataframe via describe and dtypes method to better understand the data before getting into our cleaning and manipulation steps
v. Reduce the dataset by removing independent variables that have little/no relation to our dependent variable of interest. This will ensure that our analysis is accurate and that we are only using significant variables as our predictors
vi. Ensure that any missing variables found are imputed with significant values via descriptive statistics such as mean. This keeps the integrity of the dataset.
vii. Ensure that outliers are removed that are greater than 3 standard deviations away from the mean. This will cancel out any noise and make our data more reliable
viii. Utilize dummy variables for our binary categorical values from “yes/no” to “0/1”. This will change the data type from string to numerical which is necessary for our multiple regression analysis
ix. Observe both univariate and bivariate graphical visualizations
x. The final prepared data will be saved as Log_Churn_Clean

```
#importing the necessary libraries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

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

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

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


#uploading our initial churn dataset into the pandas dataframe
churn_clean_df =
pd.read_csv(r"C:\Users\andre\OneDrive\Desktop\churn_clean.csv")


#re-labeling the Survey Questions as to make it more meaningful instead of the
generic 1-7 labling originally done
churn_clean_df.rename (columns = {'Item1': 'Timely_Response',
'Item2':'Timely_Fixes', 'Item3':'Timely_Replacements',

'Item4':'Reliability','Item5':'Options','Item6':'Respectful_Response','Item7':'
Couteous_Exchange',
            'Item8':'Evidence_Of_Active_Listening'}, inplace =
True)


#observe the dataset and get some descriptive statistics before implementing
cleaning and manipulation
churn_clean_df.shape


churn_clean_df.describe()


#based off of observations between relationships between the independent
variables and our dependent variable "Churn",
#remove less significant columns in order to reduce dataset and make it easier
for analysis
```

```
churn_clean_df = churn_clean_df.drop (columns = ['CaseOrder',
'Customer_id','Interaction','UID','City','State','County','Zip','Lat','Lng','Po
pulation','Area','TimeZone','Marital','PaymentMethod','Job'])
```

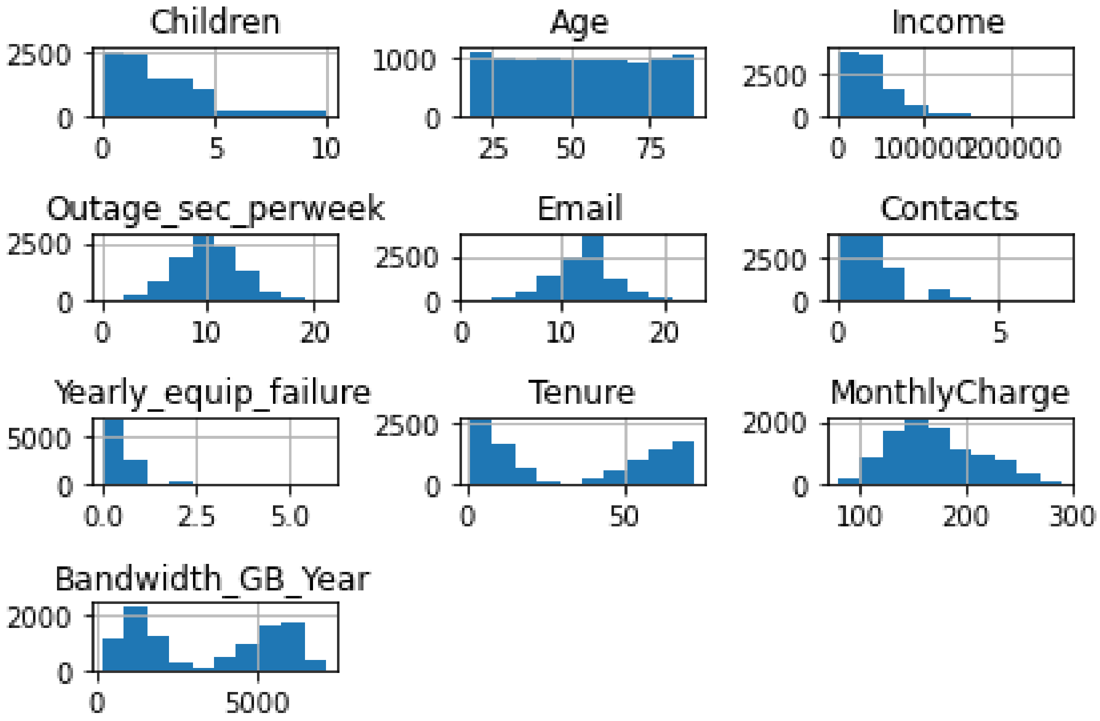
```
churn_clean_df.describe()
```

```
#addressing any missing data
missing_data_churn_df = churn_clean_df.isnull().sum()
missing_data_churn_df
```

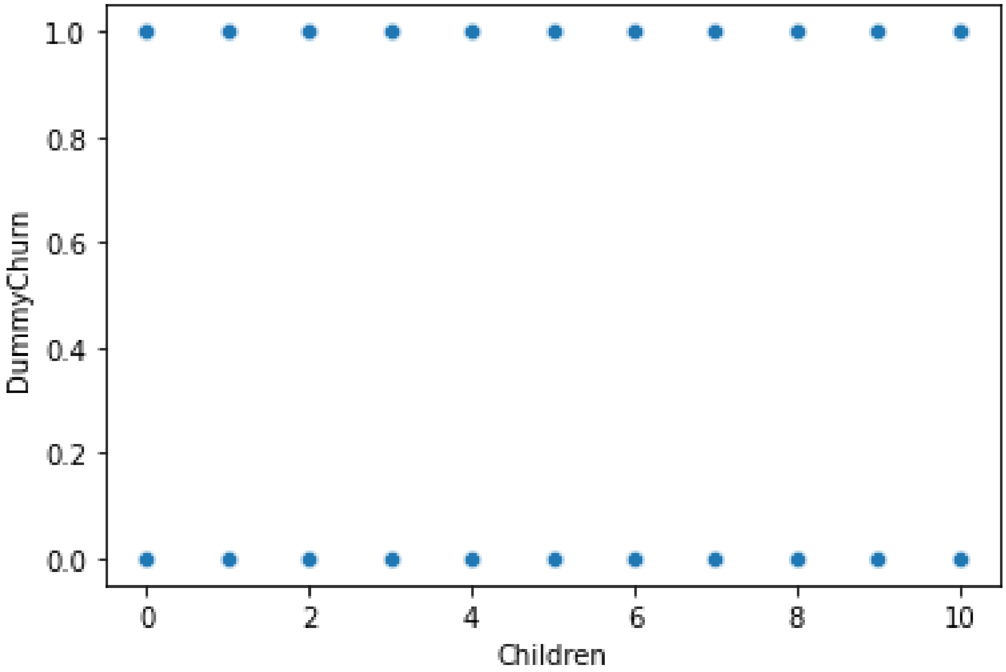
```
Children                        0
Age                             0
Income                          0
Gender                          0
Churn                           0
Outage_sec_perweek              0
Email                           0
Contacts                        0
Yearly_equip_failure            0
Techie                          0
Contract                        0
Port_modem                      0
Tablet                          0
InternetService                 0
Phone                           0
Multiple                        0
OnlineSecurity                  0
OnlineBackup                    0
DeviceProtection                0
TechSupport                     0
StreamingTV                     0
StreamingMovies                 0
PaperlessBilling                0
Tenure                          0
MonthlyCharge                   0
Bandwidth_GB_Year               0
Timely_Response                 0
Timely_Fixes                    0
Timely_Replacements             0
Reliability                     0
Options                         0
Respectful_Response             0
Couteous_Exchange               0
Evidence_Of_Active_Listening    0
dtype: int64
```

```
#we will now transform our binary categorical variables into dummy variables
taking on either a value of 0 or 1
churn_clean_df ['DummyGender'] = [1 if v =='Male' else 0 for v in
churn_clean_df['Gender']]
```

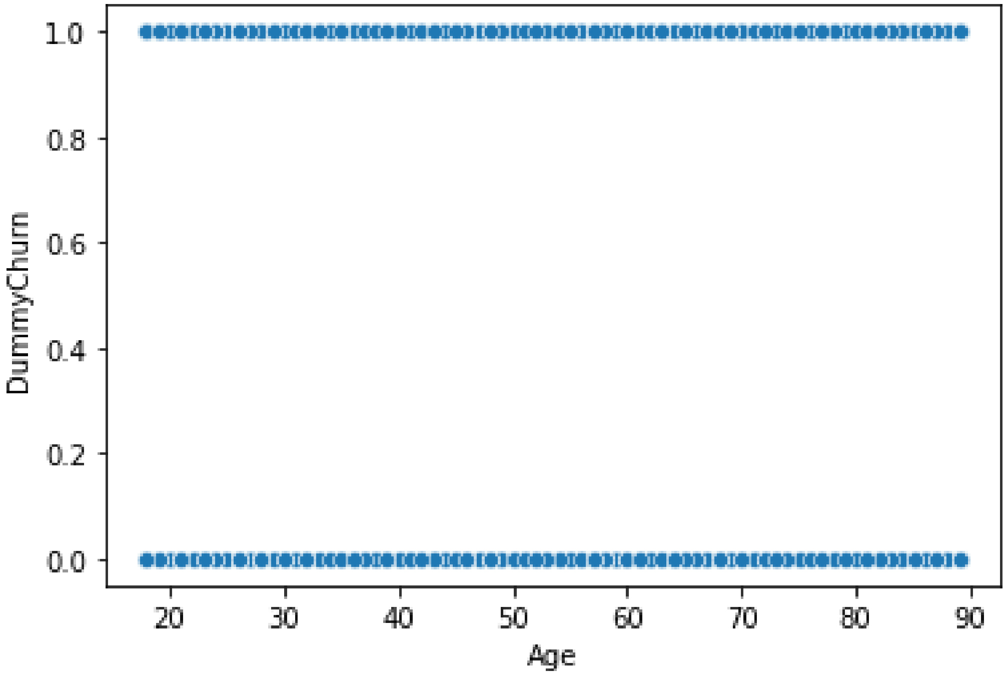
```
churn_clean_df ['DummyChurn'] = [1 if v =='Yes' else 0 for v in
churn_clean_df['Churn']]
churn_clean_df ['DummyTechie'] = [1 if v =='Yes' else 0 for v in
churn_clean_df['Techie']]
churn_clean_df ['DummyContract'] = [1 if v =='Two Year' else 0 for v in
churn_clean_df['Contract']]
churn_clean_df ['DummyPort_modem'] = [1 if v =='Yes' else 0 for v in
churn_clean_df['Port_modem']]
churn_clean_df ['DummyTablet'] = [1 if v =='Yes' else 0 for v in
churn_clean_df['Tablet']]
churn_clean_df ['DummyInternetService'] = [1 if v =='Fiber Optic' else 0 for v
in churn_clean_df['InternetService']]
churn_clean_df ['DummyPhone'] = [1 if v =='Yes' else 0 for v in
churn_clean_df['Phone']]
churn_clean_df ['DummyMultiple'] = [1 if v =='Yes' else 0 for v in
churn_clean_df['Multiple']]
churn_clean_df ['DummyOnlineSecurity'] = [1 if v =='Yes' else 0 for v in
churn_clean_df['OnlineSecurity']]
churn_clean_df ['DummyOnlineBackup'] = [1 if v =='Yes' else 0 for v in
churn_clean_df['OnlineBackup']]
churn_clean_df ['DummyDeviceProtection'] = [1 if v =='Yes' else 0 for v in
churn_clean_df['DeviceProtection']]
churn_clean_df ['DummyTechSupport'] = [1 if v =='Yes' else 0 for v in
churn_clean_df['TechSupport']]
churn_clean_df ['DummyStreamingTV'] = [1 if v =='Yes' else 0 for v in
churn_clean_df['StreamingTV']]
churn_clean_df ['DummyStreamingMovies'] = [1 if v =='Yes' else 0 for v in
churn_clean_df['StreamingMovies']]
churn_clean_df ['DummyPaperlessBilling'] = [1 if v =='Yes' else 0 for v in
churn_clean_df['PaperlessBilling']]


#now we will drop the original (Yes/No) categorical variables as we essentially
already created a duplicate of them with binary 0 or 1 values
churn_clean_df = churn_clean_df.drop (columns = ['Gender',
'Churn','Techie','Contract','Port_modem','Tablet','InternetService','Phone','Mu
ltiple','OnlineSecurity','OnlineBackup','DeviceProtection','TechSupport','Strea
mingTV','StreamingMovies','PaperlessBilling'])


#now we will double check the columns to see if we have just the dummy
variables to avoid duplications
churn_clean_df.columns


#we will use histograms for our Univariate continuous variable analysis
visualizations as they are extremely insightful into understanding the
distribution of the data

churn_clean_df[['Children','Age','Income','Outage_sec_perweek','Email','Contact
s','Yearly_equip_failure','Tenure','MonthlyCharge','Bandwidth_GB_Year']].hist()
plt.tight_layout()
```



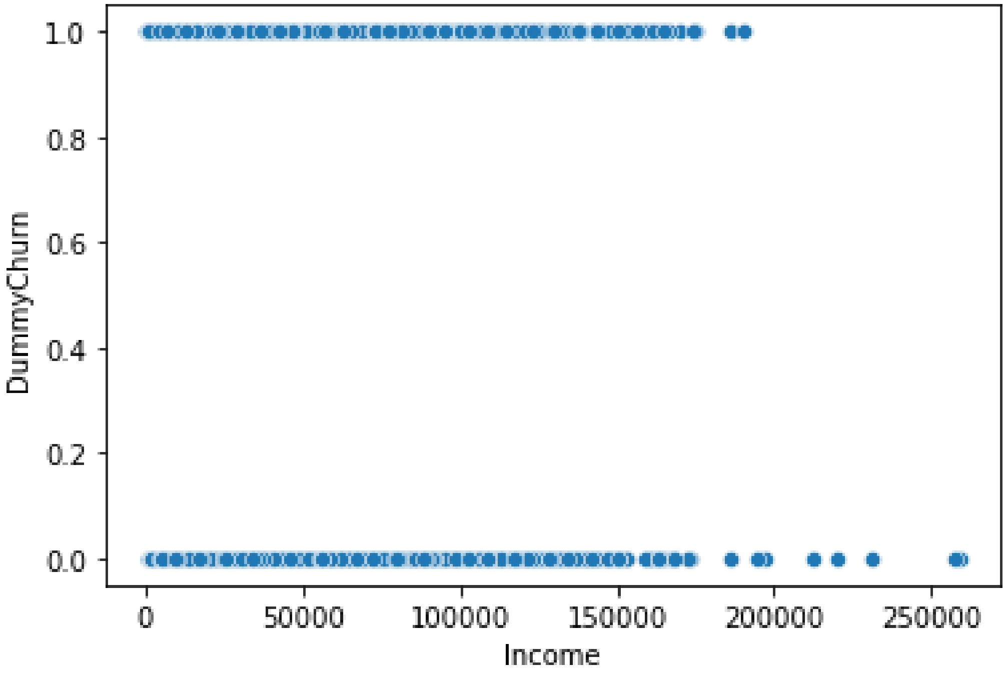
```
#Bivariate Scatter plots to observe relationship between the independent
variables and the dependent variable "Churn"
sns.scatterplot(x=churn_clean_df['Children'], y=churn_clean_df['DummyChurn'])
plt.show()
```



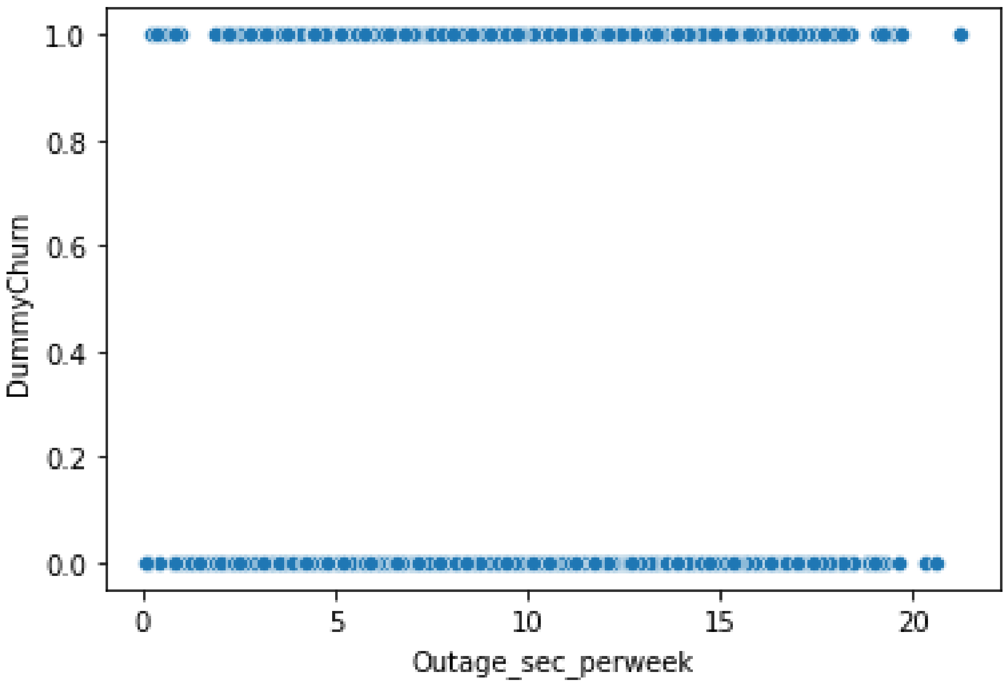
```
sns.scatterplot(x=churn_clean_df['Age'], y=churn_clean_df['DummyChurn'])
plt.show()
```



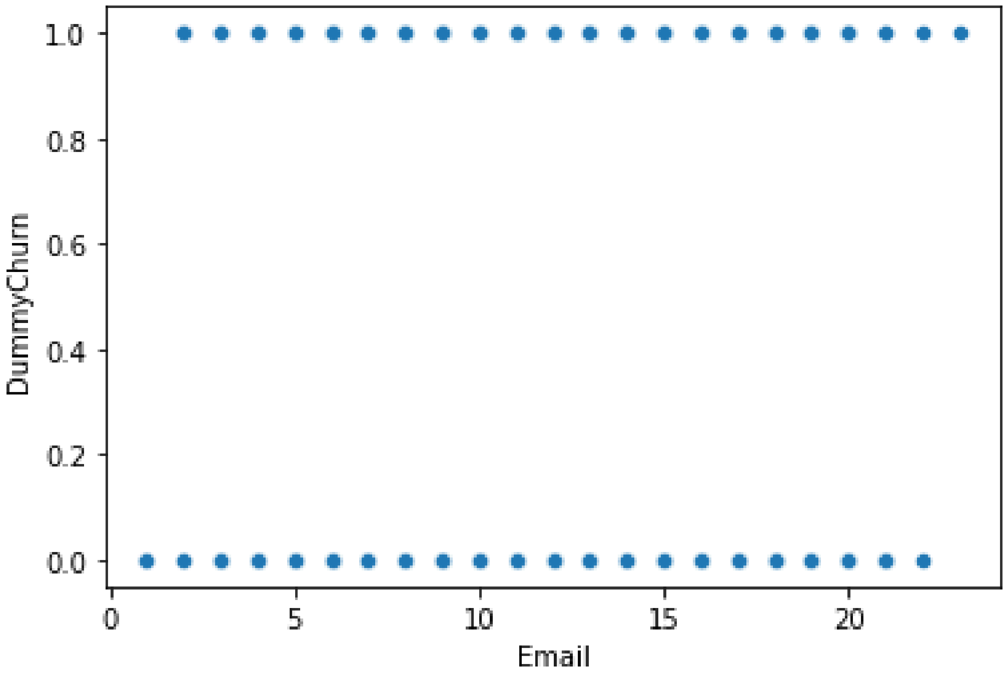
```
sns.scatterplot(x=churn_clean_df['Income'], y=churn_clean_df['DummyChurn'])
plt.show()
```



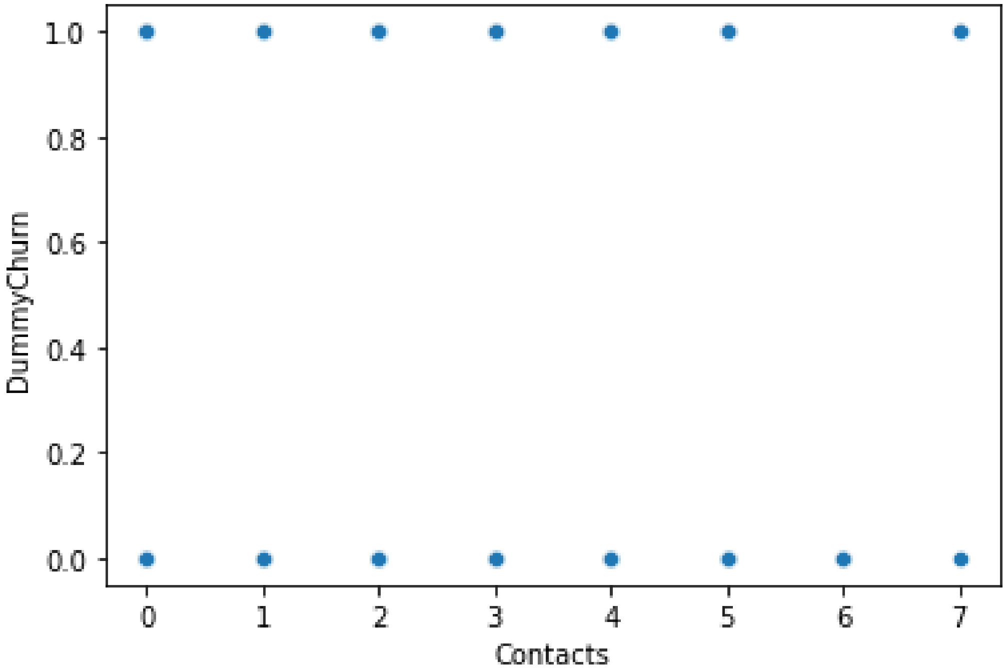
```
sns.scatterplot(x=churn_clean_df['Outage_sec_perweek'],
y=churn_clean_df['DummyChurn'])
plt.show()
```



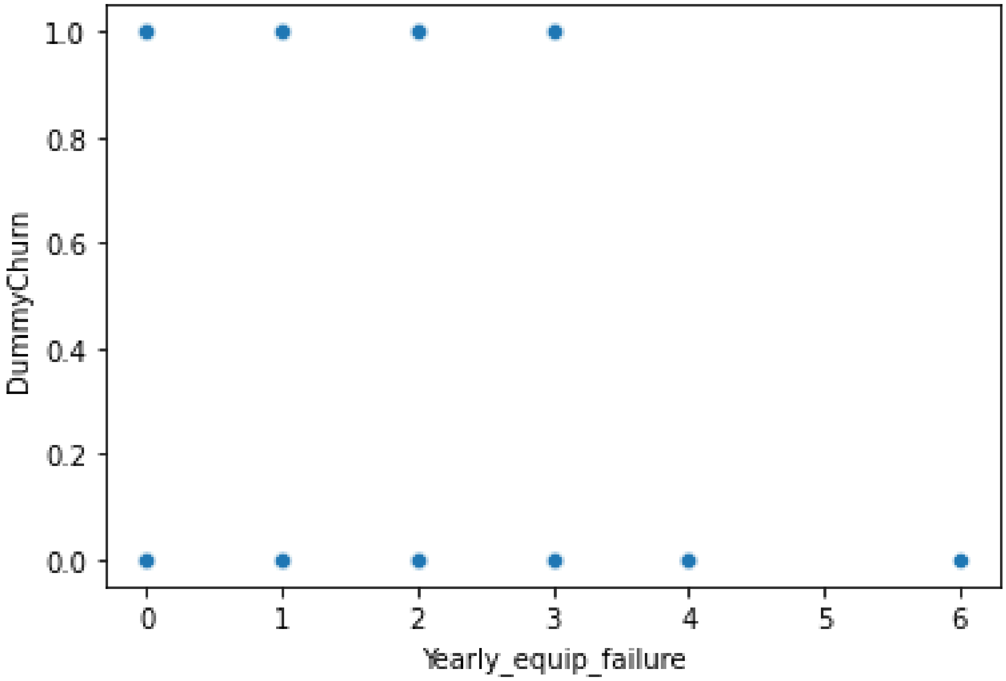
```
sns.scatterplot(x=churn_clean_df['Email'], y=churn_clean_df['DummyChurn'])
plt.show()
```



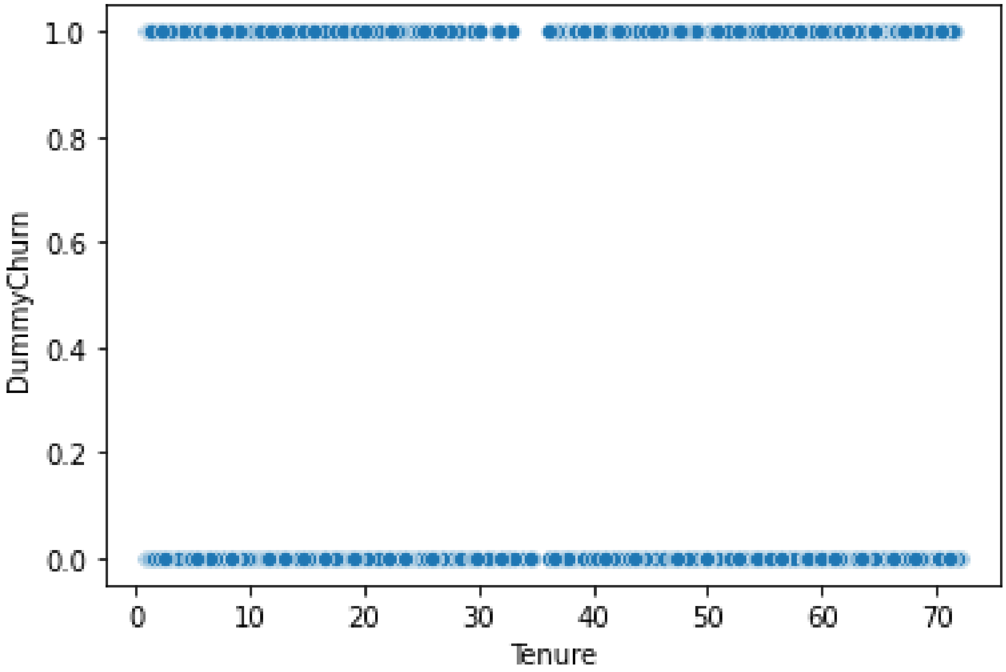
```
sns.scatterplot(x=churn_clean_df['Contacts'], y=churn_clean_df['DummyChurn'])
plt.show()
```



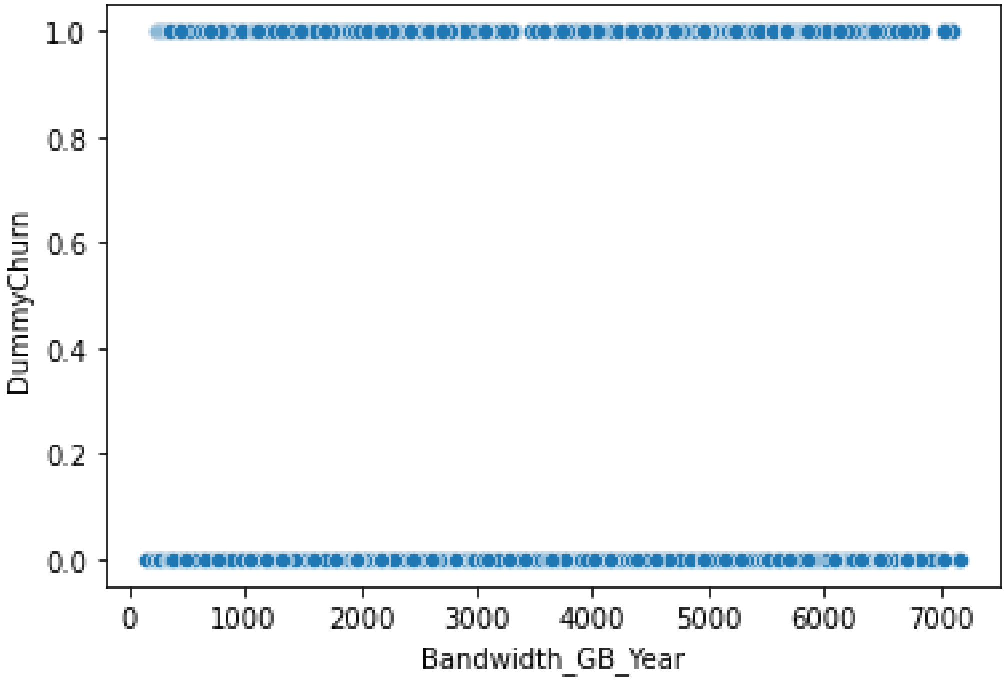
```
sns.scatterplot(x=churn_clean_df['Yearly_equip_failure'],
y=churn_clean_df['DummyChurn'])
plt.show()
```



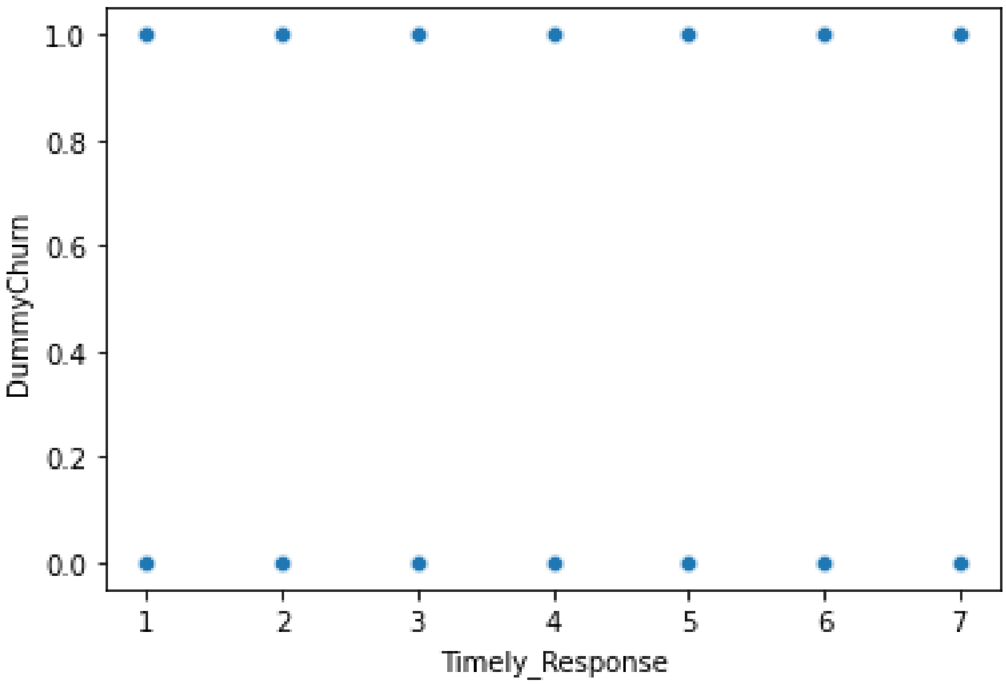
```
sns.scatterplot(x=churn_clean_df['Tenure'], y=churn_clean_df['DummyChurn'])
plt.show()
```



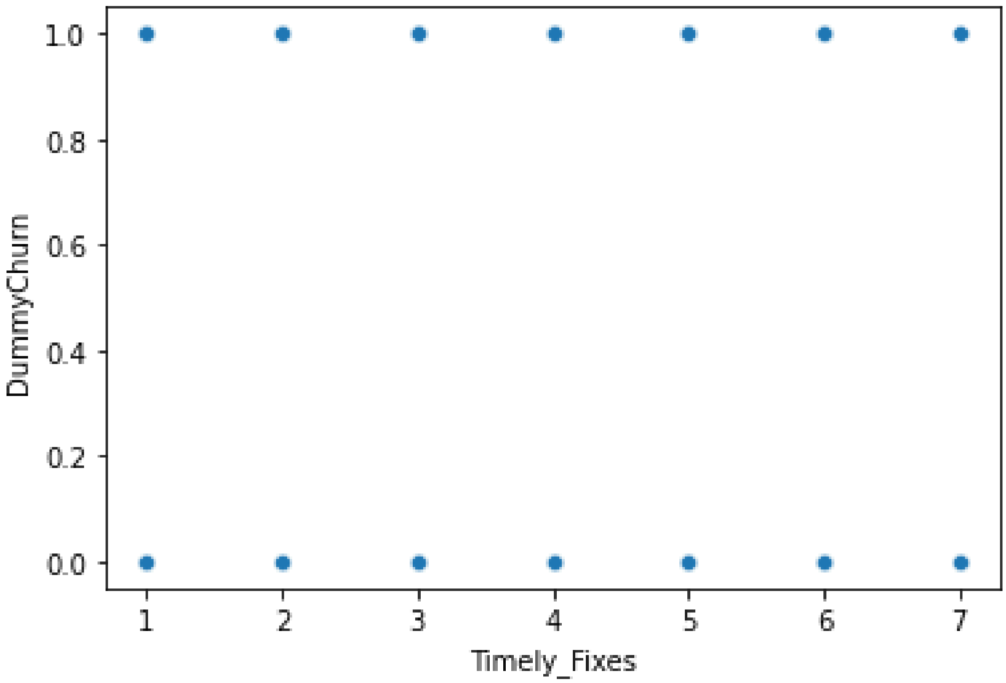
```
sns.scatterplot(x=churn_clean_df['Bandwidth_GB_Year'],
y=churn_clean_df['DummyChurn'])
plt.show()
```



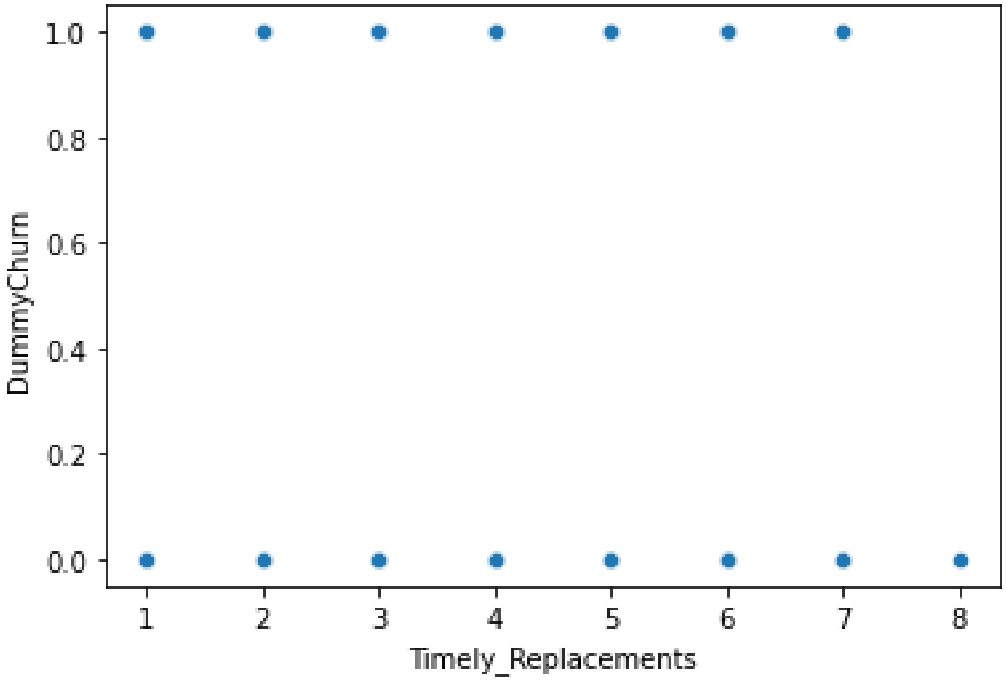
```
sns.scatterplot(x=churn_clean_df['Timely_Response'],
y=churn_clean_df['DummyChurn'])
plt.show()
```



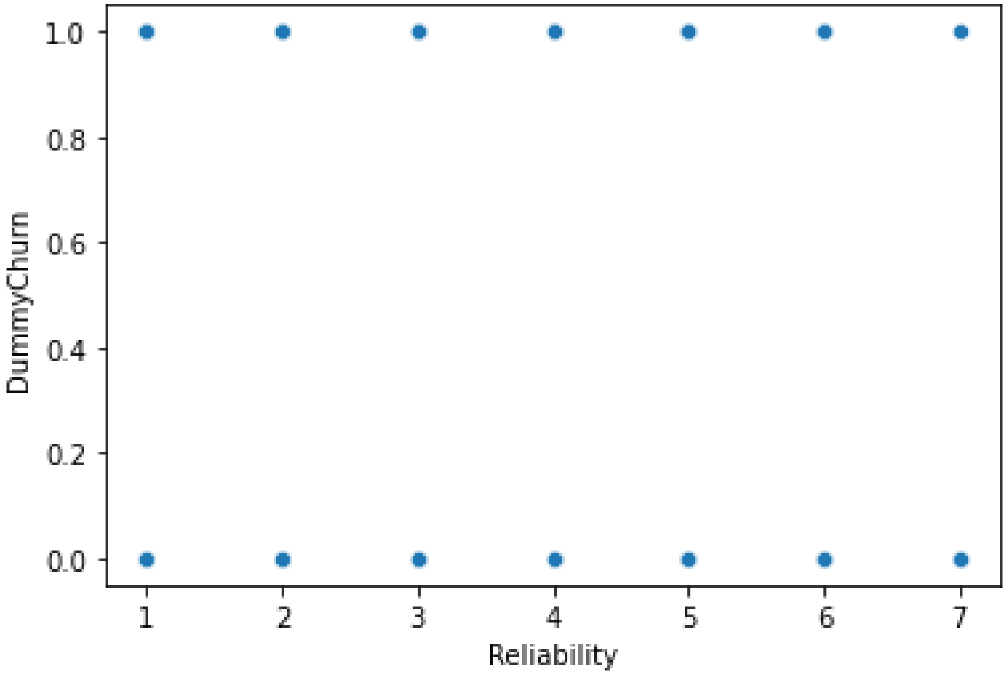
```
sns.scatterplot(x=churn_clean_df['Timely_Fixes'],
y=churn_clean_df['DummyChurn'])
plt.show()
```



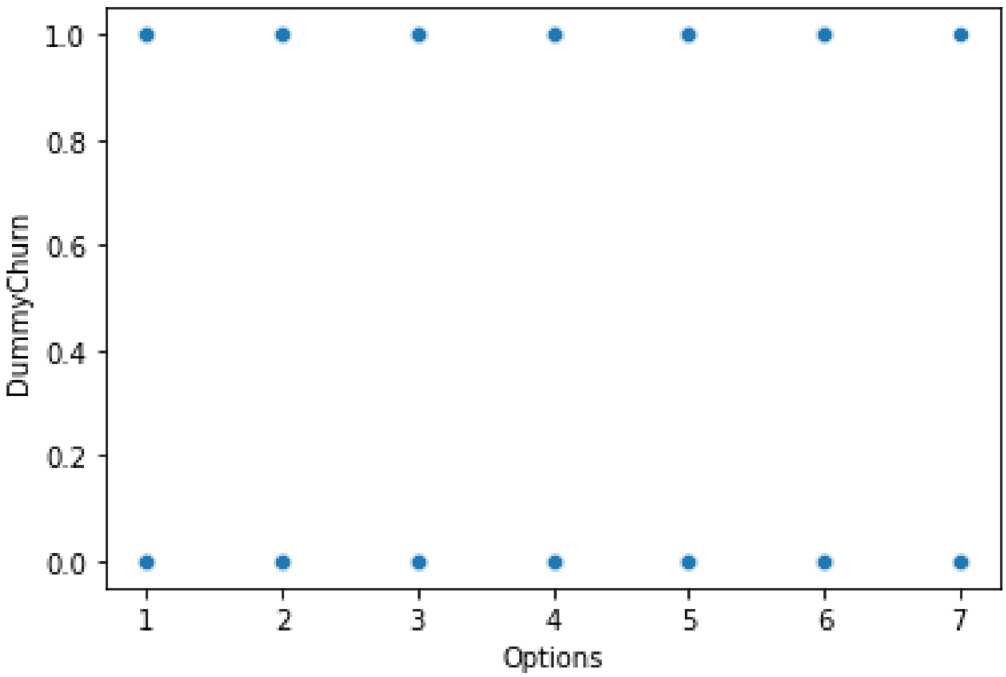
```
sns.scatterplot(x=churn_clean_df['Timely_Replacements'],
y=churn_clean_df['DummyChurn'])
plt.show()
```



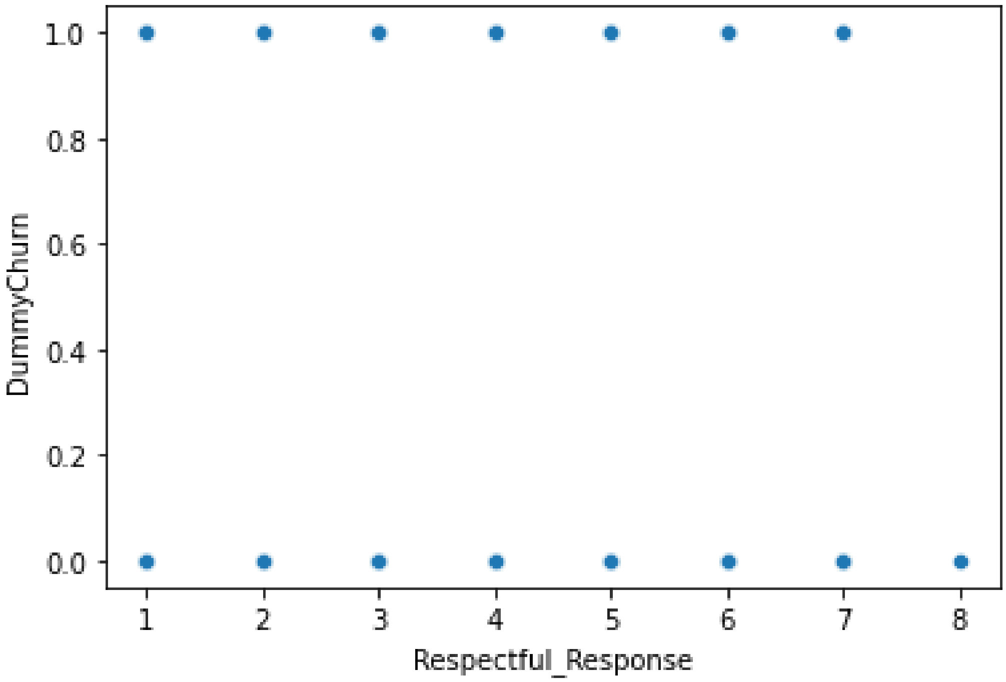
```
sns.scatterplot(x=churn_clean_df['Reliability'],
y=churn_clean_df['DummyChurn'])
plt.show()
```



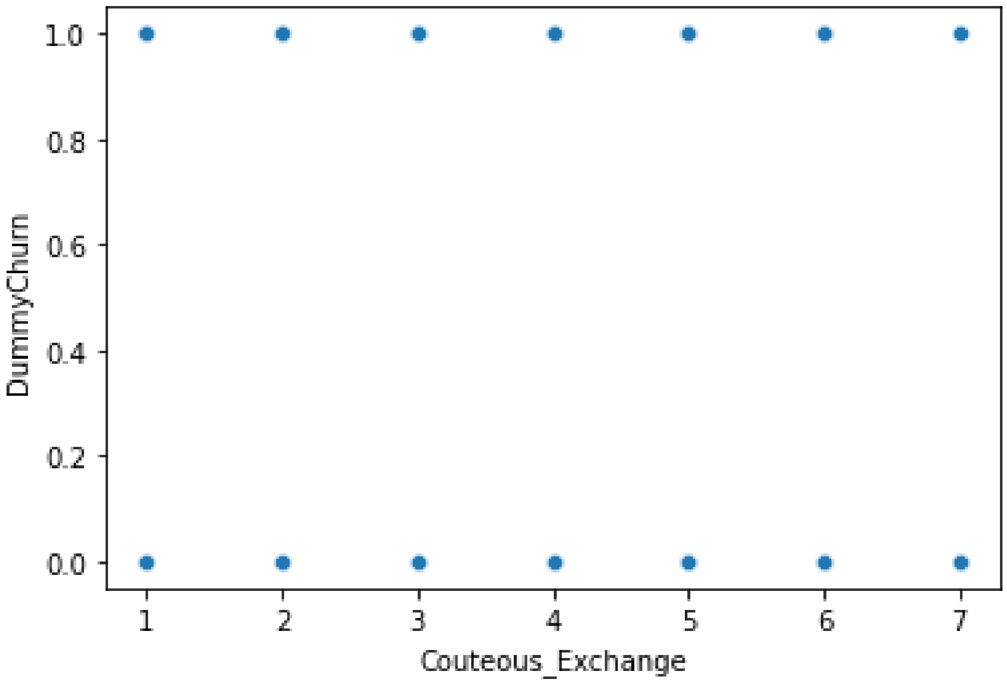
```
sns.scatterplot(x=churn_clean_df['Options'], y=churn_clean_df['DummyChurn'])
plt.show()
```



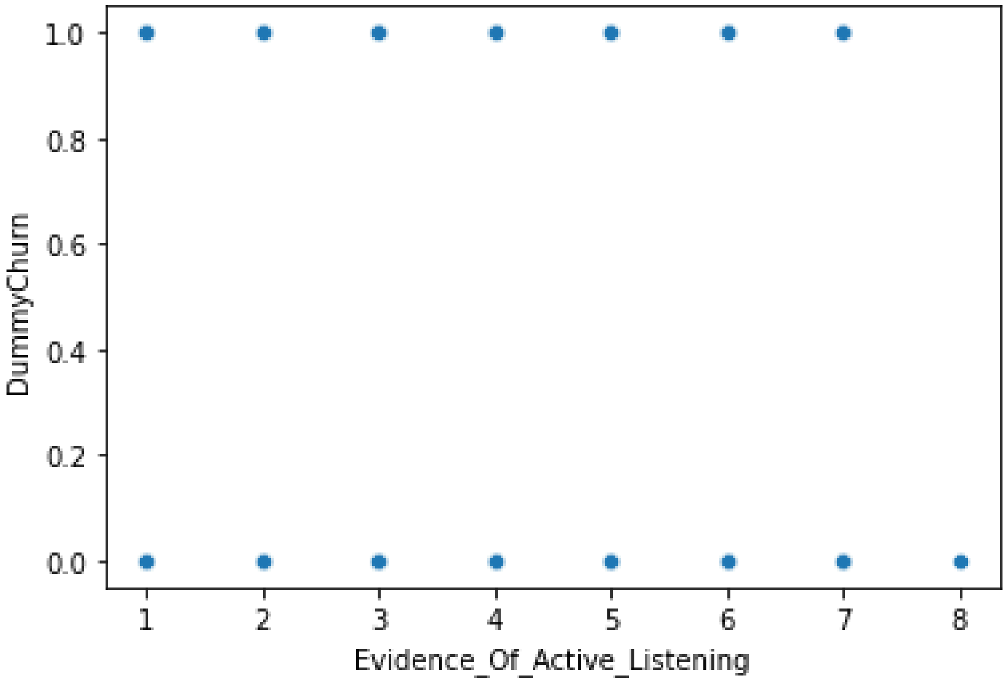
```
sns.scatterplot(x=churn_clean_df['Respectful_Response'],
y=churn_clean_df['DummyChurn'])
plt.show()
```



```
sns.scatterplot(x=churn_clean_df['Couteous_Exchange'],
y=churn_clean_df['DummyChurn'])
plt.show()
```



```
sns.scatterplot(x=churn_clean_df['Evidence_Of_Active_Listening'],
y=churn_clean_df['DummyChurn'])
plt.show()
```



```
#we will now prepare the cleaned data set for multiple regression analysis
churn_clean_df.to_csv('log_churn_clean.csv')

#use the final churn clean data from now on, so upload it into the dataframe
log_churn_clean_df = pd.read_csv('log_churn_clean.csv')
```

```
log_churn_clean_df.describe()
```

Part IV: Model Comparison & Analysis

D1. Initial Logistic Regression from all Predictors

```
#Initial Logistic Regression Model with All continuous and dummy coded
variables
log_churn_clean_df['intercept']= 1
churn_logistic = sm.Logit(log_churn_clean_df['DummyChurn'],
log_churn_clean_df[['Children','Age','Income','Outage_sec_perweek','Email','Con
tacts','Yearly_equip_failure','Tenure','Bandwidth_GB_Year','DummyGender','Month
lyCharge',
       'DummyTechie', 'DummyContract', 'DummyPort_modem',
       'DummyTablet', 'DummyInternetService', 'DummyPhone', 'DummyMultiple',
       'DummyOnlineSecurity', 'DummyOnlineBackup', 'DummyDeviceProtection',
       'DummyTechSupport', 'DummyStreamingTV', 'DummyStreamingMovies',
       'DummyPaperlessBilling','Timely_Response','Timely_Fixes',
'Timely_Replacements',

'Reliability','Options','Respectful_Response','Couteous_Exchange',
                        'Evidence_Of_Active_Listening','intercept']]).fit()
print(churn_logistic.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.270966
         Iterations 8
                           Logit Regression Results
==========================================================================
====
Dep. Variable:             DummyChurn   No. Observations:            10000
Model:                          Logit   Df Residuals:
9966
Method:                           MLE   Df Model:
33
Date:                Tue, 27 Jul 2021   Pseudo R-squ.:              0.5314
Time:                        19:03:29   Log-Likelihood:            -2709.7
converged:                       True   LL-Null:                   -5782.2
Covariance Type:            nonrobust   LLR p-value:                 0.000
==========================================================================
======================
                                  coef    std err          z      P>|z|
[0.025      0.975]
--------------------------------------------------------------------------
----------------------
Children                       -0.0504      0.018     -2.770      0.006
-0.086      -0.015
Age                             0.0082      0.002      4.208      0.000
0.004       0.012
Income                       2.976e-07   1.22e-06      0.243      0.808
-2.1e-06     2.7e-06
Outage_sec_perweek              0.0006      0.012      0.048      0.962
-0.022       0.023
```

```
Email                           -0.0018      0.011     -0.155      0.877
-0.024       0.021
Contacts                         0.0289      0.035      0.835      0.404
-0.039       0.097
Yearly_equip_failure            -0.0333      0.054     -0.613      0.540
-0.140       0.073
Tenure                          -0.2354      0.024     -9.791      0.000
-0.283      -0.188
Bandwidth_GB_Year                0.0017      0.000      5.893      0.000
0.001       0.002
DummyGender                      0.1092      0.071      1.535      0.125
-0.030       0.249
MonthlyCharge                    0.0290      0.005      6.112      0.000
0.020       0.038
DummyTechie                      0.8157      0.089      9.117      0.000
0.640       0.991
DummyContract                   -2.2879      0.103    -22.247      0.000
-2.490      -2.086
DummyPort_modem                  0.1536      0.069      2.235      0.025
0.019       0.288
DummyTablet                     -0.0752      0.075     -1.008      0.313
-0.222       0.071
DummyInternetService            -0.9108      0.188     -4.834      0.000
-1.280      -0.542
DummyPhone                      -0.3291      0.117     -2.811      0.005
-0.559      -0.100
DummyMultiple                    0.2553      0.159      1.610      0.107
-0.055       0.566
DummyOnlineSecurity             -0.3132      0.074     -4.230      0.000
-0.458      -0.168
DummyOnlineBackup               -0.1576      0.114     -1.378      0.168
-0.382       0.067
DummyDeviceProtection           -0.2319      0.084     -2.770      0.006
-0.396      -0.068
DummyTechSupport                -0.1220      0.093     -1.317      0.188
-0.304       0.060
DummyStreamingTV                 0.6961      0.185      3.762      0.000
0.333       1.059
DummyStreamingMovies             0.9203      0.228      4.034      0.000
0.473       1.368
DummyPaperlessBilling            0.1127      0.070      1.613      0.107
-0.024       0.250
Timely_Response                 -0.0176      0.049     -0.360      0.719
-0.113       0.078
Timely_Fixes                     0.0217      0.046      0.470      0.639
-0.069       0.112
Timely_Replacements             -0.0182      0.042     -0.433      0.665
-0.101       0.064
Reliability                     -0.0201      0.037     -0.539      0.590
-0.093       0.053
Options                         -0.0301      0.039     -0.770      0.441
-0.107       0.046
Respectful_Response             -0.0344      0.040     -0.861      0.389
-0.113       0.044
```

```
Couteous_Exchange                  0.0054      0.038      0.140      0.888
-0.069       0.080
Evidence_Of_Active_Listening      -0.0082      0.036     -0.228      0.819
-0.079       0.063
intercept                         -4.8763      0.498     -9.788      0.000
-5.853      -3.900
==================================================================================
====================
```

D2. Justify Selection Procedure for Reducing Initial Logistic Model

In the following step, we will be removing any insignificant variables and only leaving the significant ones as we only want to be dealing with meaningful data in our analysis. We will be doing this by removing any P-values from the variable output calculated higher than our chosen alpha of 0.05. If the P-Value is above 0.05, it is safe to assume that they are insignificant.

Once the model is reduced and re-run in our logistic analysis, we will see that the pseudo R^2 value of 0.53 will have very little change if any, meaning that we have successfully removed the insignificant values that add no further value. **(Minitab 2019).**

D3. Reduced Multiple Regression Model

```
#Reduced Logistic Regression Model with All continuous and dummy coded
variables of p-value less than alpha 0.05
log_churn_clean_df['intercept']= 1
churn_logistic_Reduced = sm.Logit(log_churn_clean_df['DummyChurn'],
log_churn_clean_df[['Children','Age','Tenure','Bandwidth_GB_Year','MonthlyCharg
e',
       'DummyTechie', 'DummyContract', 'DummyPort_modem',
'DummyInternetService', 'DummyPhone',
       'DummyOnlineSecurity', 'DummyDeviceProtection', 'DummyStreamingTV',
'DummyStreamingMovies','intercept']]).fit()
print(churn_logistic_Reduced.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.272552
         Iterations 8
                           Logit Regression Results
==============================================================================
====
Dep. Variable:            DummyChurn   No. Observations:                10000
Model:                         Logit   Df Residuals:                     9985
Method:                          MLE   Df Model:                           14
Date:               Fri, 30 Jul 2021   Pseudo R-squ.:                  0.5286
Time:                       12:20:37   Log-Likelihood:                -2725.5
converged:                      True   LL-Null:                       -5782.2
Covariance Type:           nonrobust   LLR p-value:                     0.000
==============================================================================
===============
                           coef    std err          z      P>|z|   [0.025
0.975]
---------------------------------------------------------------------------
---------------
Children                -0.0500      0.018     -2.845      0.004   -0.084
-0.016
Age                      0.0081      0.002      4.343      0.000    0.004
0.012
Tenure                  -0.2338      0.020    -11.446      0.000   -0.274
-0.194
Bandwidth_GB_Year        0.0017      0.000      6.968      0.000    0.001
0.002
MonthlyCharge            0.0306      0.002     16.374      0.000    0.027
0.034
DummyTechie              0.8057      0.089      9.063      0.000    0.631
0.980
DummyContract           -2.2737      0.103    -22.169      0.000   -2.475
-2.073
DummyPort_modem          0.1496      0.068      2.187      0.029    0.016
0.284
DummyInternetService    -0.9430      0.120     -7.879      0.000   -1.178
-0.708
DummyPhone              -0.3328      0.116     -2.869      0.004   -0.560
-0.105
DummyOnlineSecurity     -0.3240      0.074     -4.401      0.000   -0.468
-0.180
DummyDeviceProtection   -0.2427      0.071     -3.399      0.001   -0.383
-0.103
DummyStreamingTV         0.6210      0.094      6.573      0.000    0.436
0.806
DummyStreamingMovies     0.8254      0.102      8.053      0.000    0.625
1.026
intercept               -5.2651      0.251    -20.962      0.000   -5.757
-4.773
===========================================================================
===============
```

E1. Comparing the Initial and Reduced Logistic Regression Models

Here we find that even after removing 18 variables, with only 15 remaining the R^2 value of the model was only reduced by 0.0028 from 0.5316 to 0.5286. It further emphasizes that the variables with P-values above the 0.05 cut-off were indeed insignificant.

The most statistically significant variables were those with a P-value of 0, which in this case were quite a few with 10 in total:

Age, Tenure, Bandwidth_GB_Year, MonthlyCharge, DummyTechie,DummyContract, DummyInternetService, DummyOnlineSecurity, DummyStreamingTV, and DummyStreamingMovies

Our New Reduced Multiple Regression Model Equation is as follows:

Y = -5.2651- 0.0500*Children +0.0081*Age – 0.2338*Tenure + 0.0017*Bandwidth_GB_Year
+0.0306*MonthlyCharge+0.8057*DummyTechie- 2.2737 *DummyContract +
0.1496*DummyPort_Modem -0.9430*DummyInternetService
-0.3328*DummyPhone-0.3240*DummyOnlineSecurity-0.2427*DeviceProtection+
0.6210*DummyStreamingTV + 0.8254*StreamingMovies

E2. Output and Calculations of the Analysis

Confusion Matrix output and calculations are provided below and all other outputs and calculations are provided above

```
#Prepare and create a confusion matrix
data = churn_clean_df
X = data.iloc[:, 1:-1].values
y = data.iloc[:, -1].values

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.3,
random_state = 0)

from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression (random_state=0)
classifier.fit(X_train, y_train)

y_predict = classifier.predict(X_test)

from sklearn.metrics import confusion_matrix
confusion_matrix = confusion_matrix(y_test, y_predict)
print(confusion_matrix)
```

```
[[   0 1251]
 [   0 1749]]
```

```
from sklearn.model_selection import cross_val_score
accuracy = cross_val_score (estimator = classifier, X=X_train, y=y_train,
cv=10)
print("accuracy:{:.2f}%".format(accuracy.mean()*100))
print("Standard Deviation: {:.2f}%".format(accuracy.std()*100))
```

```
accuracy:59.04%
Standard Deviation: 0.07%
```

```
#Classification Report
from sklearn.metrics import classification_report
print(classification_report(y_test, y_predict))
```

```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00      1251
           1       0.58      1.00      0.74      1749
```

```
    accuracy                                  0.58      3000
   macro avg       0.29      0.50      0.37      3000
weighted avg       0.34      0.58      0.43      3000
```

E3. Code of Multiple Logistic Analysis

Code provided above

Part V: Data Summary and Implications

F1. Results of Data Analysis

The regression equation for the reduced model is as follows

Y = -5.2651- 0.0500*Children +0.0081*Age – 0.2338*Tenure + 0.0017*Bandwidth_GB_Year +0.0306*MonthlyCharge+0.8057*DummyTechie- 2.2737 *DummyContract + 0.1496*DummyPort_Modem -0.9430*DummyInternetService -0.3328*DummyPhone-0.3240*DummyOnlineSecurity-0.2427*DeviceProtection+ 0.6210*DummyStreamingTV + 0.8254*StreamingMovies

It is important to note that there is a lot of positive as well as negative coefficients in the reduced model output. This means that there are some clear positive as well as inverse relationships between the variables and the churn result. Since the premise of the business question is focusing on reducing customer churn, we should pay attention to the negative coefficient variables as these are the ones that seem to reduce customer churn as they increase.

In this case they are the following: Tenure, Children, Contract, InternetService, Phone, OnlineSecurity, and DeviceProtection.

As far as limitations the data set that we have of 10,000 observations is abit small, and so perhaps if we focus more efforts on collection of data to increase the observations, we will have a more accurate prediction of the relationships.

F2. Recommended Course of Action

It is first important to look at the negative relationships between the churn outcome and the variables in our analysis. As listed above, the inverse relationship tend to stem from a place of service, safety, and duration (explained by variables such as Tenure, InternetService, and OnlineSecurity/DeviceProtection).

Therefore, as a company, they should ensure to allocate resources towards giving customers the best possible contracts, as well as grant them access to fast and high speed internet, with adequate software security. The more these increase in unit value, the more customer churn will decrease by the respective coefficient unit value explained in the above logistic output.

This makes sense as these variables tend to be the primary aspects of general usage by customers from a telecom company. Finally, if these is not all achievable, I would recommend that atleast in the short run, the company focus on the "Contract" as it is something that they can make policies fairly quickly to put into place. This is a good move as it is also the variable with the highest negative weight, which can equate to the biggest opportunity in reducing customer churn.

G. Panapto Video

https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=d588ee0a-6e96-4aba-9b7e-ad77016489aa

H. Third Party Code Sources

Avinash Navlani (2019). Understanding Logistic Regression in Python. https://www.datacamp.com/community/tutorials/understanding-logistic-regression-python

Terence Shin (2020). Understanding the Confusion Matrix and How To Implement It In Python. https://towardsdatascience.com/understanding-the-confusion-matrix-and-how-to-implement-it-in-python-319202e0fe4d

I. Acknowledged Sources

Minitab (2019). Model Reduction. https://support.minitab.com/en-us/minitab-express/1/help-and-how-to/modeling-statistics/regression/supporting-topics/regression-models/model-reduction/

Zendesk (2021). How To Calculate Customer Churn Rate. https://www.zendesk.com/blog/customer-churn-rate/

Enoch Kan (2018). Data Science 101: Is Python Better Than R? https://towardsdatascience.com/data-science-101-is-python-better-than-r-b8f258f57b0f

Mirko Stojiljkovic (2021). Logistic Regression in Python. https://realpython.com/logistic-regression-python/