**Jason Pena**

**WGU University**

# D208 PERFORMANCE ASSESSMENT NBM2 TASK 2

# LOGISTIC REGRESSION FOR PREDICTIVE MODELING

**Part I: Research Question**

A.  Describe the purpose of this data analysis by doing the following:

1.  Summarize **one** research question that is relevant to a real-world organizational situation captured in the data set you have selected and that you will answer using multiple regression.

Which customers are at high risk of churn? And is it possible to determine which features are most significant to churn?

2.  Define the objectives or goals of the data analysis. Ensure that your objectives or goals are reasonable within the scope of the data dictionary and are represented in the available data.

The goal or objective of this analysis is to be able to predict in advance that a specific customer is likely to churn. "The churn rate, also known as the rate of attrition, is the rate at which customers stop doing business with an entity. It is most expressed as the percentage of service subscribers who discontinue their subscriptions within a given time period (Investopedia, 2020) Stakeholders in the company can benefit by this analysis by understanding more effectively which customers are likely to churn soon because this will provide weight for decisions in marketing improved services to customers with these characteristics and past user experiences.

**Part II: Method Justification**

B.  Describe logistic regression methods by doing the following:

1.  Summarize the assumptions of a logistic regression model.

“First, binary logistic regression requires the dependent variable to be binary and ordinal logistic regression require the dependent variable to be ordinal.” (Statisticssolutions.com, 2021).

“Second, logistic regression requires the observations to be independent of each other. In other words, the observations should not come from repeated measurements or matched data. (Statisticssolutions.com, 2021).

“Third, logistic regression requires there to be little or no multicollinearity among the independent variables. This means that the independent variables should not be too highly correlated to each other.” (Statisticssolutions.com, 2021).

“Fourth, logistic regression assumes linearity of independent variables and log odds.” (Statisticssolutions.com, 2021).

2.  Describe the benefits of using the tool(s) you have chosen (i.e., Python, R, or both) in support of various phases of the analysis.

Python will be used to support the various phases of the analysis. Python will allow the user to implement coding solutions, manipulating the data, and creating visual representations for the performance assessment.

3.  Explain why multiple regression is an appropriate technique to analyze the research question summarized in Part I.

“Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between dependent binary variable and one or more nominal, ordinal, interval, or ratio-level independent variables. (Statisticssolutions.com, 2021).

Logistic regression is an appropriate technique to analyze the research question because the dependent variable is binomial, “Yes” or “No”. The result is to find out what the likelihood of customer churn is for individual customer, based on a list of independent variables (job, children, age, income, etc.) This will create a stronger understanding of the increased probability of churn as the analysis will input different independent variables to find out whether or not they have a positive or negative relationship to the target variable.

Part III: Data Preparation

C.Summarize the data preparation process for logistic regression by doing the following:

1. Describe your data preparation goals and the data manipulations that will be used to achieve the goals.

The data preparation goals and data manipulations will include:

1. Import the dataset into Python.
2. Evaluate the data structure to gain a better understanding of the variables and data types.
3. Provide a name to identify my dataset. The naming convention I chose for my dataset is: Churn\_df
4. Data manipulations made to the data set will be named: df
5. Check for any misleading variable names and rename them.
6. Check for any missing data that could skew the model.
7. Missing data will be inputted with measures of central tendency.
8. Create visualizations to identify any outliers that could affect the model.
9. Summaries of univariate and bivariate statistics to search for any flags

2. Discuss the summary statistics, including the target variable and all predictors that you will need to gather from the data set to answer the research question.

Once the data is imported into Python, the output provides that the dataset consists of 50 original columns and 10,000 records. For purposes of this analysis, columns such as User ID, & demographic variables such as (Caseorder, Customer\_id, Interaction, UID, City, State, County, Zip, Lat, Lng, Population, Area, TimeZone, Job, Marial, and PaymentMethod will be removed from the dataframe. Binary categorical variables were encoded to 1 / 0 as well as the ordinal variables being converted into numeric. After analyzing summary statistics, it appears that the dataset has been sufficiently cleaned leaving no null, NAs, or missing data points.

This results in having 34 remaining numerical independent predictor variables including the target variable. The most vital step to the decision-making process is the dependent variable of “Churn” which is binary categorical with only two values, “Yes” or “No”. “Churn” will be the categorical target variable.

Measures of central tendency through histograms & boxplots revealed normal distributions for "Email", "MonthlyCharge" and "Outage\_sec\_perweek". The cleaned dataset no longer retained any outliers. Histograms for "Bandwidth\_GB\_Year" & "Tenure" displayed a bimodal distributions, which demonstrated a direct linear relationship in a scatterplot. The average customer was 53 years-old (with a standard deviation of 20 years), had 2 children (with a standard deviation of 2 kids), an income of 39,806 (with a standard deviation of about 30,000), experienced 10 outage-seconds/week, was marketed to by email 12 times, contacted technical support less than one time, had less than 1 yearly equipment failure, has been with the company for 34.5 months, has a monthly charge of approximately 173 & uses 3,392 GBs/year.

During the exploratory data analysis phase, we may discover relevance in the continuous predictor variables:

* Age (Age of customer as reported in sign-up information)
* Bandwidth\_GB\_Year (Data usage in gigabytes)
* Children (Number of children)
* Contacts (Number of times customer contacted technical support)
* Email (Number of emails sent out to customer)
* Income (Annual income of customer)
* MonthlyCharge (The amount charged to the customer monthly)
* Outage\_sec\_perweek (Average number of seconds per week of system outages in the customer’s neighborhood)
* Tenure (Number of months the customer has stayed with the provider)
* Yearly\_equip\_failure (The number of times customer’s equipment failed and had to be reset/replaced in the past year)

There also could be relevance in the categorical predictor variables (all binary categorical with only two values, “Yes” or “No”, ordinal variables will be noted in order):

* Gender (Customer self-identification as male, female, or nonbinary)
* Techie: (Whether the customer considers themselves technically inclined (based on customer questionnaire when they signed up for services) (yes, no))
* Contract: (The contract term of the customer (month-to-month, one year, two year))
* Port\_modem: (Whether the customer has a portable modem (yes, no))
* Tablet: (Whether the customer owns a tablet such as iPad, Surface, etc. (yes, no))
* InternetService: (Customer’s internet service provider (DSL, fiber optic, None))
* Phone: (Whether the customer has a phone service (yes, no))
* Multiple: (Whether the customer has multiple lines (yes, no))
* OnlineSecurity: (Whether the customer has an online security add-on (yes, no))
* OnlineBackup: (Whether the customer has an online backup add-on (yes, no))
* DeviceProtection: (Whether the customer has device protection add-on (yes, no))
* TechSupport: (Whether the customer has a technical support add-on (yes, no))
* StreamingTV: (Whether the customer has streaming TV (yes, no))
* StreamingMovies: (Whether the customer has streaming movies (yes, no))
* PaperlessBilling: (Whether the customer has paperless billing (yes, no))

The discrete ordinal predictor variables from the survey responses regarding various customer features may be relevant in the decision-making process. In the surveys, customers provided ordinal numerical data by rating 8 customer service factors on a scale of 1 to 8 (1 = most important, 8 = least important).

* Item1: Timely response
* Item2: Timely fixes
* Item3: Timely replacements
* Item4: Reliability
* Item5: Options
* Item6: Respectful response
* Item7: Courteous exchange
* Item8: Evidence of active listening

3. Explain the steps used to prepare the data for the analysis, including the annotated code.

Annotated Code with explanation of each step:

# Standard Data Science Imports

import numpy as np

import pandas as pd

from pandas import Series, DataFrame

Here I am importing NumPy as pd and pandas as pd. I am also importing Series, and DataFrame

“NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.” (Numpy.org, 2021)

“Similar to Numpy, pandas deals primarily with data in 1-D and 2-D arrays; however, pandas handles them differently.“ (Educative.io, 2021)

“In pandas, 1-D arrays are referred to a series. A **series** is created through the pd.Series constructor, which has a lot of optional arguments. The most common argument is data, which specifies the elements of the series.” (Educative.io, 2021)

“A DataFrame is simply a 2-D array. It can be created through the pd.DataFrame constructor, which takes in essentially the same arguments as pd.Series. However, while a series could be constructed from a scalar (representing a single value Series), a DataFrame cannot.” (Educative.io, 2021)

# Visualization libraries

import seaborn as sns

import matplotlib.pyplot as plt

%matplotlib inline

Next I will be importing the Visualization libraries.

Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures.

Seaborn helps you explore and understand your data. Its plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots. Its dataset-oriented, declarative API lets you focus on what the different elements of your plots mean, rather than on the details of how to draw them. (Seaborn.pydata.org, 2021)

# Statistics packages

import pylab

from statsmodels.formula.api import logit

import statistics

from scipy import stats

After, are the Statistics packages

PyLab is a procedural interface to the Matplotlib object-oriented plotting library. Matplotlib is the whole package; matplotlib.pyplot is a module in Matplotlib; and PyLab is a module that gets installed alongside Matplotlib. PyLab is a convenience module that bulk imports matplotlib.pyplot (for plotting) and NumPy (for Mathematics and working with arrays) in a single name space. (tutorialspoints.com, 2021)

# Scikit-learn

import sklearn

from sklearn import preprocessing

from sklearn.preprocessing import LabelEncoder

from sklearn import metrics

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import confusion\_matrix

from sklearn.model\_selection import cross\_val\_score

from sklearn.metrics import classification\_report

Then I will now import the Scikit-learn

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering, and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib. (tutorialspoints.com, 2021)

# Load data set into Pandas Dataframe

churn\_df = pd.read\_csv(r'C:\Users\Hydraconix\Desktop\churn\_clean.csv')

Here I am importing the dataset using Pandas. Once I have imported the data, I will check the description of dataframe, structure (columns & rows) & data types.

# Checking for Null Values

churn\_df.isna().sum()

Output:

CaseOrder 0

Customer\_id 0

Interaction 0

UID 0

City 0

State 0

County 0

Zip 0

Lat 0

Lng 0

Population 0

Area 0

TimeZone 0

Job 0

Children 0

Age 0

Income 0

Marital 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

PaymentMethod 0

Tenure 0

MonthlyCharge 0

Bandwidth\_GB\_Year 0

Item1 0

Item2 0

Item3 0

Item4 0

Item5 0

Item6 0

Item7 0

Item8 0

dtype: int64

Here I am checking for any missing values. If any were found, they would be replaced with dummy variables using one of the central tendencies. (Mean, Median, Mode)

# Rename Last 8 Survey Columns for better description of variables

churn\_df.rename(columns = {'Item1' : 'TimelyResponse',

'Item2' : 'Fixes' ,

'Item3' : 'Replacements' ,

'Item4' : 'Reliability' ,

'Item5' : 'Options' ,

'Item6' : 'Respectfulness' ,

'Item7' : 'Courteous' ,

'Item8' : 'Listening'},

inplace=True)

In this step I am renaming columns/variables of survey to easily recognizable features (ex: “Item1” will be renamed “TimelyResponse”).

#Summary Statistics

churn\_df.Age.describe()

churn\_df.Children.describe()

churn\_df.Income.describe()

churn\_df.Outage\_sec\_perweek.describe()

churn\_df.Yearly\_equip\_failure.describe()

churn\_df.Tenure.describe()

churn\_df.MonthlyCharge.describe()

churn\_df.Bandwidth\_GB\_Year.describe()

# Remove less meaningful demographic variables from statistics description

churn\_df = churn.drop(columns=['Caseorder' ,

'Customer\_id' ,

'Interaction' ,

'UID' ,

'City' ,

'State' ,

'County' ,

'Zip' ,

'Lat' ,

'Lng' ,

'Population' ,

'Area' ,

'TimeZone' ,

'Job' ,

'Marital' ,

'PaymentMethod'])

Drop less meaningful identifiers (ex: “Customer ID”) & demographic columns (ex: zip code) from dataframe.

# Converting binary categorical variables to numeric variables

churn\_df['DummyChurn'] = [1 if v == 'Yes' else 0 for v in churn\_df['Churn']]

churn\_df['DummyTechie'] = [1 if v == 'Yes' else 0 for v in churn\_df['Techie']]

churn\_df['DummyPort\_modem'] = [1 if v == 'Yes' else 0 for v in churn\_df['Port\_modem']]

churn\_df['DummyTablet'] = [1 if v == 'Yes' else 0 for v in churn\_df['Tablet']]

churn\_df['DummyPhone'] = [1 if v == 'Yes' else 0 for v in churn\_df['Phone']]

churn\_df['DummyMultiple'] = [1 if v == 'Yes' else 0 for v in churn\_df['Multiple']]

churn\_df['DummyOnlineSecurity'] = [1 if v == 'Yes' else 0 for v in churn\_df['OnlineSecurity']]

churn\_df['DummyOnlineBackup'] = [1 if v == 'Yes' else 0 for v in churn\_df['OnlineBackup']]

churn\_df['DummyDeviceProtection'] = [1 if v == 'Yes' else 0 for v in churn\_df['DeviceProtection']]

churn\_df['DummyTechSupport'] = [1 if v == 'Yes' else 0 for v in churn\_df['TechSupport']]

churn\_df['DummyStreamingTV'] = [1 if v == 'Yes' else 0 for v in churn\_df['StreamingTV']]

churn\_df['DummyStreamingMovies'] = [1 if v == 'Yes' else 0 for v in churn\_df['StreamingMovies']]

churn\_df['DummyPaperlessBilling'] = [1 if v == 'Yes' else 0 for v in churn\_df['PaperlessBilling']]

Created dummy variables to encode categorical, yes/no data points into 1/0 numerical values.

# Converting ordinal categorical data into numeric variables

churn\_df['DummyInternetService'] = churn\_df.InternetService.map({'None' : 0, 'DSL' : 1, 'Fiber Optic' : 2})

churn\_df['DummyContract'] = churn\_df.InternetService.map({'Month-to-month' : 0, 'One year' : 1, 'Two Year' : 2})

churn\_df['DummyGender'] = churn\_df.Gender.map({'Nonbinary' : 0, 'Male' : 1, 'Female' : 2})

Created dummy variables to encode ordinal categorical data, ordinal data points into 0,1,2 numerical values.

# Drop original categorical features from dataframe

churn\_df = churn\_df.drop(columns=['Gender' ,

'Churn' ,

'Techie' ,

'Contract' ,

'Port\_modem' ,

'Tablet' ,

'InternetService' ,

'Phone' ,

'Multiple' ,

'OnlineSecurity' ,

'OnlineBackup',

'DeviceProtection' ,

'TechSupport' ,

'StreamingTV',

'StreamingMovies',

'PaperlessBilling'])

Removing original categorical features from dataframe

4. Generate univariate and bivariate visualizations of the distributions of variables in the cleaned data set. Include the target variable in your bivariate visualizations.

Univariate Statistics

#Create histograms of continuous variables

churn\_df[['Children',

'Age' ,

'Income' ,

'Outage\_sec\_perweek' ,

'Email' ,

'Contacts' ,

'Yearly\_equip\_failure' ,

'Tenure' , 'MonthlyCharge' ,

'Bandwidth\_GB\_Year' ,

'DummyGender' ,

'DummyInternetService' ,

'DummyContract']].hist()

plt.savefig('churn\_pyplot.jpg')

plt.tight\_layout()

Graphical user interface, diagram, application

Description automatically generated

# Create Seaborn Boxplots for continuous variables

sns.boxplot('Tenure' , data = churn\_df)

plt.show()

Chart

Description automatically generated

sns.boxplot('MonthlyCharge' , data = churn\_df)

plt.show()

Chart

Description automatically generated

sns.boxplot('Bandwidth\_GB\_Year' , data = churn\_df)

plt.show()

Chart, histogram

Description automatically generated

It appears that anomalies have been removed from the dataset present "churn\_clean.csv" as there are no remaining outliers.

Bivariate Statistics

I will now run some scatterplots to get an idea of our linear relationships with our target variable of “DummyChurn”

# Run Scatterplots to show direct or inverse relationships between the target & independent variables

# Run scatterplots to show direct or inverse relationships between target & independent variables

sns.scatterplot(x=churn\_df['Children'], y=churn\_df['DummyChurn'], color='red')

plt.show()

Chart, scatter chart

Description automatically generated

sns.scatterplot(x=churn\_df['Age'], y=churn\_df['DummyChurn'], color='red')

plt.show()

A picture containing chart

Description automatically generated

sns.scatterplot(x=churn\_df['Income'], y=churn\_df['DummyChurn'], color='red')

plt.show()

Graphical user interface, text

Description automatically generated

sns.scatterplot(x=churn\_df['DummyGender'], y=churn\_df['DummyChurn'], color='red')

plt.show()

Chart, scatter chart

Description automatically generated

sns.scatterplot(x=churn\_df['Outage\_sec\_perweek'], y=churn\_df['DummyChurn'], color='red')

plt.show()

Graphical user interface

Description automatically generated with low confidence

sns.scatterplot(x=churn\_df['Email'], y=churn\_df['DummyChurn'], color='red')

plt.show()

Chart, histogram

Description automatically generated with medium confidence

sns.scatterplot(x=churn\_df['Contacts'], y=churn\_df['DummyChurn'], color='red')

plt.show()

Chart, scatter chart

Description automatically generated

sns.scatterplot(x=churn\_df['Yearly\_equip\_failure'], y=churn\_df['DummyChurn'], color='red')

plt.show()

Chart, scatter chart

Description automatically generated

sns.scatterplot(x=churn\_df['DummyTechie'], y=churn\_df['DummyChurn'], color='red')

plt.show()

Chart

Description automatically generated

sns.scatterplot(x=churn\_df['Tenure'], y=churn\_df['DummyChurn'], color='red')

plt.show()

A picture containing graphical user interface

Description automatically generated

sns.scatterplot(x=churn\_df['MonthlyCharge'], y=churn\_df['DummyChurn'], color='red')

plt.show()

A picture containing text

Description automatically generated

sns.scatterplot(x=churn\_df['Bandwidth\_GB\_Year'], y=churn\_df['DummyChurn'], color='red')

plt.show()

Graphical user interface, text

Description automatically generated

sns.scatterplot(x=churn\_df['TimelyResponse'], y=churn\_df['DummyChurn'], color='red')

plt.show()

Chart, scatter chart

Description automatically generated

sns.scatterplot(x=churn\_df['Fixes'], y=churn\_df['DummyChurn'], color='red')

plt.show()

Chart, scatter chart

Description automatically generated

sns.scatterplot(x=churn\_df['Replacements'], y=churn\_df['DummyChurn'], color='red')

plt.show()

Chart, scatter chart

Description automatically generated

sns.scatterplot(x=churn\_df['Reliability'], y=churn\_df['DummyChurn'], color='red')

plt.show()

Chart, scatter chart

Description automatically generated

sns.scatterplot(x=churn\_df['Options'], y=churn\_df['DummyChurn'], color='red')

plt.show()

Chart, scatter chart

Description automatically generated

sns.scatterplot(x=churn\_df['Respectfulness'], y=churn\_df['DummyChurn'], color='red')

plt.show()

Chart, scatter chart

Description automatically generated

sns.scatterplot(x=churn\_df['Courteous'], y=churn\_df['DummyChurn'], color='red')

plt.show()

Chart, scatter chart

Description automatically generated

sns.scatterplot(x=churn\_df['Listening'], y=churn\_df['DummyChurn'], color='red')

plt.show()

Chart, scatter chart

Description automatically generated

These scatterplots suggest no correlation between the “DummyChurn” & any of our continuous variables or categorical variables.

5. Provide a copy of the prepared data set.

churn\_df.to\_csv('churn\_prepared\_log.csv')

Part IV: Model Comparison and Analysis

D. Compare an initial and a reduced logistic regression model by doing the following:

1. Construct an initial logistic regression model from all predictors that were identified in Part C2.

# Construct an initial logistic regression model from all predictors that were identified in Part C2. This first regression model is using the continuous variables that were given in the data set.

churn\_logit\_model = logit("DummyChurn ~ Children + Age + Income + Outage\_sec\_perweek + Email +Contacts +Yearly\_equip\_failure +Tenure + MonthlyCharge + TimelyResponse + Fixes + Replacements + Reliability + Options + Respectfulness + Courteous + Listening", data=churn\_df).fit()

print(churn\_logit\_model.params)

print(churn\_logit\_model.summary())

Output:

Intercept -4.765275e+00

Children -8.099363e-03

Age 1.601289e-03

Income 7.464227e-07

Outage\_sec\_perweek -1.522054e-03

Email 1.807264e-03

Contacts 2.866639e-02

Yearly\_equip\_failure -3.554854e-02

Tenure -7.391671e-02

MonthlyCharge 3.320066e-02

TimelyResponse -3.590255e-02

Fixes -8.073790e-03

Replacements -2.210886e-03

Reliability -4.007931e-02

Options -3.692864e-02

Respectfulness -2.182938e-03

Courteous -2.239337e-02

Listening 1.947222e-03

dtype: float64

Logit Regression Results

===============================================================

Dep. Variable: DummyChurn No. Observations: 10000

Model: Logit Df Residuals: 9982

Method: MLE Df Model: 17

Date: Mon, 29 Nov 2021 Pseudo R-squ.: 0.4092

Time: 19:48:51 Log-Likelihood: -3416.4

converged: True LL-Null: -5782.2

Covariance Type: nonrobust LLR p-value: 0.000

===============================================================

coef std err z P>|z| [0.025 0.975]

----------------------------------------------------------------------------------------

Intercept -4.7653 0.353 -13.513 0.000 -5.456 -4.074

Children -0.0081 0.014 -0.565 0.572 -0.036 0.020

Age 0.0016 0.001 1.091 0.275 -0.001 0.004

Income 7.464e-07 1.08e-06 0.693 0.489 -1.37e-06 2.86e-06

Outage\_sec\_perweek -0.0015 0.010 -0.149 0.881 -0.021 0.018

Email 0.0018 0.010 0.179 0.858 -0.018 0.022

Contacts 0.0287 0.031 0.939 0.348 -0.031 0.089

Yearly\_equip\_failure -0.0355 0.048 -0.739 0.460 -0.130 0.059

Tenure -0.0739 0.002 -41.754 0.000 -0.077 -0.070

MonthlyCharge 0.0332 0.001 37.097 0.000 0.031 0.035

TimelyResponse -0.0359 0.043 -0.828 0.408 -0.121 0.049

Fixes -0.0081 0.041 -0.197 0.843 -0.088 0.072

Replacements -0.0022 0.037 -0.059 0.953 -0.075 0.071

Reliability -0.0401 0.033 -1.209 0.227 -0.105 0.025

Options -0.0369 0.035 -1.063 0.288 -0.105 0.031

Respectfulness -0.0022 0.035 -0.062 0.951 -0.072 0.067

Courteous -0.0224 0.034 -0.663 0.508 -0.089 0.044

Listening 0.0019 0.032 0.061 0.952 -0.061 0.065

===============================================================

### Now, let's run a model including all encoded categorical dummy variables.

churn\_logit\_model2 = logit("DummyChurn ~ Children + Age + Income + Outage\_sec\_perweek + Email +Contacts +Yearly\_equip\_failure +Tenure + MonthlyCharge + TimelyResponse + Fixes + Replacements + Reliability + Options + Respectfulness + Courteous + Listening + Bandwidth\_GB\_Year + DummyTechie + DummyPort\_modem + DummyTablet + DummyPhone + DummyMultiple + DummyOnlineSecurity + DummyOnlineBackup + DummyDeviceProtection + DummyTechSupport + DummyStreamingTV + DummyStreamingMovies + DummyPaperlessBilling + DummyInternetService + DummyContract + DummyGender", data=churn\_df).fit()

print(churn\_logit\_model2.params)

print(churn\_logit\_model2.summary())

Output:

Intercept -5.232259e+00

Children -9.022241e-02

Age 1.259720e-02

Income 2.802104e-07

Outage\_sec\_perweek 2.083996e-03

Email -5.992616e-03

Contacts 4.472942e-02

Yearly\_equip\_failure -3.463630e-02

Tenure -3.927217e-01

MonthlyCharge 3.156403e-02

TimelyResponse -2.209419e-02

Fixes 1.466799e-02

Replacements 4.312380e-03

Reliability -1.847220e-02

Options -2.831939e-02

Respectfulness -3.108024e-02

Courteous 8.629632e-03

Listening -1.311182e-02

Bandwidth\_GB\_Year 3.446072e-03

DummyTechie 1.003803e+00

DummyPort\_modem 1.547360e-01

DummyTablet -6.078450e-02

DummyPhone -3.109035e-01

DummyMultiple 2.830334e-01

DummyOnlineSecurity -4.439935e-01

DummyOnlineBackup -2.576343e-01

DummyDeviceProtection -3.171974e-01

DummyTechSupport -1.315971e-01

DummyStreamingTV 6.390119e-01

DummyStreamingMovies 9.108723e-01

DummyPaperlessBilling 1.561046e-01

DummyInternetService -4.899604e-01

DummyContract -1.938148e+00

DummyGender -3.877672e-02

dtype: float64

Logit Regression Results

===============================================================

Dep. Variable: DummyChurn No. Observations: 10000

Model: Logit Df Residuals: 9966

Method: MLE Df Model: 33

Date: Mon, 29 Nov 2021 Pseudo R-squ.: 0.6009

Time: 20:57:20 Log-Likelihood: -2307.6

converged: True LL-Null: -5782.2

Covariance Type: nonrobust LLR p-value: 0.000

===============================================================

coef std err z P>|z| [0.025 0.975]

-----------------------------------------------------------------------------------------

Intercept -5.2323 0.549 -9.535 0.000 -6.308 -4.157

Children -0.0902 0.018 -4.899 0.000 -0.126 -0.054

Age 0.0126 0.002 6.436 0.000 0.009 0.016

Income 2.802e-07 1.33e-06 0.211 0.833 -2.33e-06 2.89e-06

Outage\_sec\_perweek 0.0021 0.013 0.166 0.868 -0.023 0.027

Email -0.0060 0.012 -0.486 0.627 -0.030 0.018

Contacts 0.0447 0.037 1.193 0.233 -0.029 0.118

Yearly\_equip\_failure -0.0346 0.059 -0.587 0.557 -0.150 0.081

Tenure -0.3927 0.018 -22.249 0.000 -0.427 -0.358

MonthlyCharge 0.0316 0.005 6.905 0.000 0.023 0.041

TimelyResponse -0.0221 0.053 -0.418 0.676 -0.126 0.081

Fixes 0.0147 0.050 0.292 0.770 -0.084 0.113

Replacements 0.0043 0.046 0.095 0.925 -0.085 0.094

Reliability -0.0185 0.040 -0.456 0.648 -0.098 0.061

Options -0.0283 0.043 -0.664 0.507 -0.112 0.055

Respectfulness -0.0311 0.044 -0.712 0.476 -0.117 0.054

Courteous 0.0086 0.042 0.207 0.836 -0.073 0.090

Listening -0.0131 0.039 -0.335 0.737 -0.090 0.063

Bandwidth\_GB\_Year 0.0034 0.000 17.162 0.000 0.003 0.004

DummyTechie 1.0038 0.098 10.211 0.000 0.811 1.196

DummyPort\_modem 0.1547 0.075 2.069 0.039 0.008 0.301

DummyTablet -0.0608 0.081 -0.746 0.456 -0.220 0.099

DummyPhone -0.3109 0.128 -2.428 0.015 -0.562 -0.060

DummyMultiple 0.2830 0.169 1.679 0.093 -0.047 0.613

DummyOnlineSecurity -0.4440 0.081 -5.508 0.000 -0.602 -0.286

DummyOnlineBackup -0.2576 0.129 -2.002 0.045 -0.510 -0.005

DummyDeviceProtection -0.3172 0.096 -3.294 0.001 -0.506 -0.128

DummyTechSupport -0.1316 0.097 -1.362 0.173 -0.321 0.058

DummyStreamingTV 0.6390 0.217 2.939 0.003 0.213 1.065

DummyStreamingMovies 0.9109 0.257 3.543 0.000 0.407 1.415

DummyPaperlessBilling 0.1561 0.076 2.052 0.040 0.007 0.305

DummyInternetService -0.4900 0.091 -5.392 0.000 -0.668 -0.312

DummyContract -1.9381 0.064 -30.265 0.000 -2.064 -1.813

DummyGender -0.0388 0.069 -0.560 0.576 -0.175 0.097

===============================================================

**Initial Multiple Linear Regression Model**

With *30* independent variables (17 continuous & 13 categorical):  
  
 y = -5.23 - 0.0902 \* Children + 0.0126 \* Age + 2.802e-07 \* Income - 0.0021 \* Outage\_sec\_perweek - 0.0060 \* Email + 0.0447 \* Contacts - 0.0346 \* Yearly\_equip\_failure + 1.0038 \* DummyTechie - 1.938 \* DummyContract + 0.154 \* DummyPort\_modem - 0.0608 \* DummyTablet - 0.4900 \* DummyInternetService - 0.3109 \* DummyPhone - 0.2830 \* DummyMultiple - 0.4400 \* DummyOnlineSecurity - 0.2576 \* DummyOnlineBackup - 0. \* 3172 DummyDeviceProtection - 0.1316 \* DummyTechSupport + 0.6390 \* DummyStreamingTV + 0.1561 \* DummyPaperlessBilling - 0.3927 \* Tenure + 0.0316 \* MonthlyCharge - 0.0221 \* TimelyResponse + 0.0147 \* Fixes + 0.0047 \* Replacements - 0.0185 \* Reliability - 0.0283 \* Options - 0.0311 \* Respectfulness + 0.0086 \* Courteous - 0.0131 \* Listening

With the R value = 0.6009, which is not a good indication for the variance of the model. Due to not having much luck with the visualizations, the regression ended up using all of the possible predictors given in the data set. Coefficients that showed higher than 0/5 were DummyTechie, DummyInternetService, DummyContract, DummyStreamingTV. These variables also have p-values 0.00, therefore, are significant.

2. Justify a statistically based variable selection procedure and a model evaluation metric to reduce the initial model in a way that aligns with the research question.

# Create dataframe for heatmap bivariate analysis of correlation

churn\_bivariate = churn\_df[['DummyChurn', 'Children', 'Age', 'Income',

'Outage\_sec\_perweek', 'Yearly\_equip\_failure', 'DummyTechie', 'DummyContract',

'DummyPort\_modem', 'DummyTablet', 'DummyInternetService',

'DummyPhone', 'DummyMultiple', 'DummyOnlineSecurity',

'DummyOnlineBackup', 'DummyDeviceProtection',

'DummyTechSupport', 'DummyStreamingTV',

'DummyPaperlessBilling','Email', 'Contacts',

'Tenure', 'MonthlyCharge', 'Bandwidth\_GB\_Year', 'TimelyResponse', 'Fixes',

'Replacements', 'Reliability', 'Options', 'Respectfulness',

'Courteous', 'Listening']]

# Run Seaborn heatmap

sns.heatmap(churn\_bivariate.corr(), annot=False)

plt.show()

Chart

Description automatically generated

I will now remove some of the variables that were less significant to get a better understanding.

churn\_bivariate = churn\_df[['DummyChurn', 'Bandwidth\_GB\_Year', 'Children',

'Tenure', 'TimelyResponse', 'Fixes',

'Replacements', 'Respectfulness',

'Courteous', 'Listening']]

sns.heatmap(churn\_bivariate.corr(), annot=True)

plt.show()

Chart

Description automatically generated

DummyChurn does not appear to be well correlated with any of our variables any of the variables.

Once again, based on the R value = 0.6009 as well as the bivariate analysis on all of the predictor variables, it appears that most of the variables show little to no significant value with “DummyChurn”. The analysis will move forward using the variables that showed coefficients higher than 0.5 and p values that were around 0.00. The variables are DummyTechie, DummyInternetService, DummyContract and DummyStreamingTV.

3. Provide a reduced logistic regression model

# Run reduced logistic regression

churn\_logit\_model\_reduced = logit("DummyChurn ~ DummyTechie + DummyContract + DummyInternetService + DummyStreamingTV", data=churn\_df).fit()

print(churn\_logit\_model\_reduced.summary())

Output:

Optimization terminated successfully.

Current function value: 0.511366

Iterations 6

Logit Regression Results

==========================================================

Dep. Variable: DummyChurn No. Observations: 10000

Model: Logit Df Residuals: 9995

Method: MLE Df Model: 4

Date: Sun, 05 Dec 2021 Pseudo R-squ.: 0.1156

Time: 19:01:50 Log-Likelihood: -5113.7

converged: True LL-Null: -5782.2

Covariance Type: nonrobust LLR p-value: 2.972e-288

==========================================================

coef std err z P>|z| [0.025 0.975]

----------------------------------------------------------------------------------------

Intercept -1.1987 0.057 -21.146 0.000 -1.310 -1.088

DummyTechie 0.4222 0.062 6.784 0.000 0.300 0.544

DummyContract -0.8645 0.035 -24.932 0.000 -0.932 -0.797

DummyInternetService -0.0560 0.031 -1.797 0.072 -0.117 0.005

DummyStreamingTV 1.1845 0.050 23.619 0.000 1.086 1.283

==========================================================

**Reduced Logistic Regression Model**

With 4 indpendent variables:

 y = -1.1987 + 0.4222 \* DummyTechie - 0.8645 \* DummyContract - 0.0560 \* DummyInternetServices + 1.1845 \* DummyStreamingTV

E. Analyze the data set using your reduced logistic regression model by doing the following:

1. Explain your data analysis process by comparing the initial and reduced logistic regression models, including the following elements:

The logic of the variable selection technique

### Confusion Matrix

# Import the prepared dataset

dataset = pd.read\_csv('data/churn\_prepared\_log.csv')

X = dataset.iloc[:, 1:-1].values

y = dataset.iloc[:, -1].values

# Split the dataset into the Training set and Test set

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.2, random\_state = 0)

# Training the Logistic Regression model on the Training set

classifier = LogisticRegression(random\_state = 0)

classifier.fit(X\_train, y\_train)

# Predict the Test set results

y\_pred = classifier.predict(X\_test)

# Make the Confusion Matrix

cm = confusion\_matrix(y\_test, y\_pred)

print(cm)

[ 0 1 43]

[ 0 64 887]

[ 0 57 948]

y\_predict\_test = classifier.predict(X\_test)

cm2 = confusion\_matrix(y\_test, y\_predict\_test)

sns.heatmap(cm2, annot=True)

Chart

Description automatically generated with medium confidence

# Classification Report

print(classification\_report(y\_test, y\_predict\_test))

precision recall f1-score support

0 0.00 0.00 0.00 44

1 0.52 0.07 0.12 951

2 0.50 0.94 0.66 1005

accuracy 0.51 2000

macro avg 0.34 0.34 0.26 2000

weighted avg 0.50 0.51 0.39 2000

2. Provide the output and any calculations of the analysis you performed, including a confusion matrix.

Calculations & code output above.

3. Provide the code used to support the implementation of the logistic regression models.

All code for analysis is included above.

F. Summarize your findings and assumptions by doing the following:

1. Discuss the results of your data analysis, including the following elements:

* A regression equation for the reduced model

y = -1.1987 + 0.4222 \* DummyTechie - 0.8645 \* DummyContract - 0.0560 \* DummyInternetServices + 1.1845 \* DummyStreamingTV

* An interpretation of coefficients of the statistically significant variables of the model

The coefficients suggest that for every 1 unit of:

DummyTechie – DummyChurn will increase 0.4222 units

DummyContract – DummyChurn will decrease 0.8645 units

DummyInternerService – DummyChurn will decrease by 0.0560 units

DummyStreamingTV – DummyChurn will increase by 1.1945 units

* The statistical and practical significance of the model

The p-values for all variables were at 0.00 as well as their coefficients being above 0.5.

* The limitations of the analysis

The limitations of this analysis are that the data set is a bit small & that perhaps more years of data need to be collected. Also, correlation is not causation so we cannot tell whether ‘Contracts’ and the customer’s ‘Internet Service’ are variables that cause correlation. More investigation is required.

2. Recommend a course of action based on your results

Given the negative coefficients of DummyContract & DummyInternetServices, we suggest additional marketing for contracts & internet services as those with contract appear less likely to leave the company.

H. List the web sources used to acquire data or segments of third-party code to support the application. Ensure the web sources are reliable.

Logistic Regression & Sklearn kit

<https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8>

I. Acknowledge sources, using in-text citations and references, for content that is quoted, paraphrased, or summarized.

A. K., By, -, Aniruddha Kalbandehttps://www.fireblazeaischool.in/blogsOn the mission to train 1, Kalbande, A., & On the mission to train 1. (2021, July 17). *Assumptions of linear regression algorithm - blogs: Fireblaze AI School*. Blogs | Fireblaze AI School. Retrieved October 24, 2021, from <https://www.fireblazeaischool.in/blogs/assumptions-of-linear-regression/>

Assumptions of logistic regression. Statistics Solutions. (2021, August 11). Retrieved November 27, 2021, from https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/assumptions-of-logistic-regression/.

Frankenfield, J. (2021, May 19). Churn rate. Investopedia. Retrieved October 24, 2021, from <https://www.investopedia.com/terms/c/churnrate.asp>