

Andreas Sebastian Dunn

Western Governors University, D208: Predictive Modeling

September 18, 2021

D208, Task 2

A logistic regression analysis on churn data in the telecom industry

Part 1

This paper will use logistic regression to try and determine the relationship between customer churn and our collected data on telecom customers. Churn is a big deal for the telecom industry because, while retaining customers is relatively cheap, it is much more expensive to get new customers. We will use logistic regression to try and answer the following question: can we determine which customers are more likely to churn and which predictor variables are most significant in predicting churn?

The objectives of this analysis are to answer the research question about customer churn using logistic regression and the available data. I will be using the predictor variables of Gender, Income, Age, Children, Bandwidth_GB_Year, Yearly_equip_failure, Outage_sec_perweek, and MonthlyCharge.

Part 2

The assumptions of the logistic regression model are as follows: it is based on Bernoulli distribution and not Gaussian like multiple regression, the value of Y is always a categorical value (in this case, Yes or No to churn), it predicts the probability that we will find a certain outcome and not the outcome itself, and there can't be multi-collinearity among the X variables.

I will be using Python for this analysis because Python is widely used for performing regression analysis and has many libraries that focus on data science. I will be using Python libraries to also show visual distributions of the data.

Since we are trying to estimate the probability that a customer will churn, the most appropriate regression equation is logistic regression; we need to use logistic regression since our Y

value is binomial and not a number. Logistic regression gives us the ability to single-out and reduce our X variables so we can further understand the relationship between our features and our response; it will help us in determining the factors that influence our categorical variable Churn.

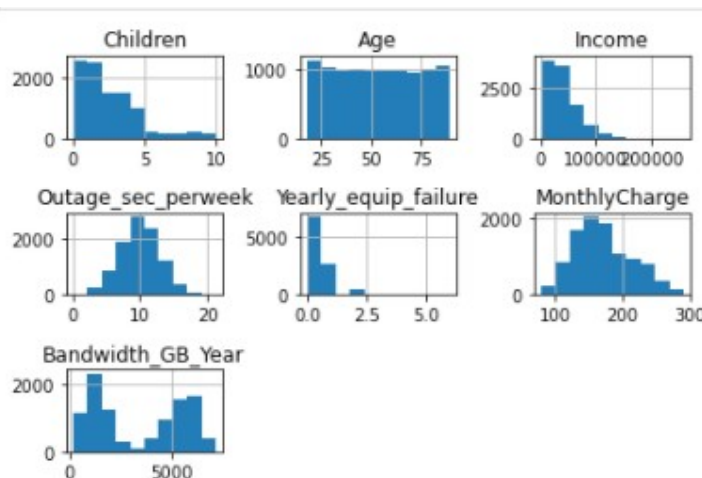
Part 3

I will first import the customer churn .csv file into Python and create a dataframe out of the data. I will check for any missing or null values and replace with the mean from the corresponding column if any are found. I will then make sure that all our data types are consistent and ready for the regression equation.

Summary statistics show that the mean is 2 for Children, 53 for Age, \$39,806 for Income, 10 for Outage_sec_perweek, 0.4 for Yearly_equip_failure, 172 for MonthlyCharge, and 3392 for Bandwidth_GB_Year. The standard deviations are as follows: 2.14 for Children, 20.7 for Age, 28199 for Income, 3 for Outage_sec_perweek, 0.63 for Yearly_equip_failure, 43 for MonthlyCharge, and 2185 for Bandwidth_GB_Year. Judging from the summary statistics alone and looking at the standard deviations, I predict that our R^2 is going to be low and our regression line will decrease in slope due to the presence of outliers in Income, Age, and Children.

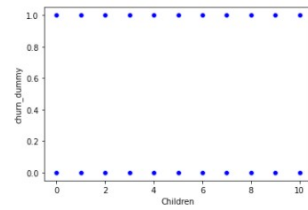
Using built in Python methods, I checked for null values and found none. Then I checked the data types and found that they are all consistent for the purposes of logistic regression. Lastly, I dropped some meaningless and demographic data because it is not relevant towards our research question and imputed the Python dataframe into the regression method.

Univariate Distributions:

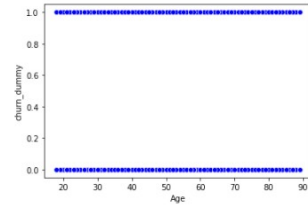


Bivariate Distributions:

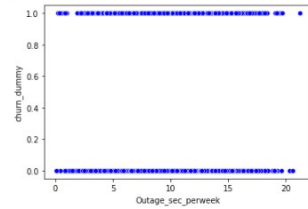
```
[8]: sns.scatterplot(x=df['Children'], y=df['churn_dummy'], color='blue')  
plt.show();
```



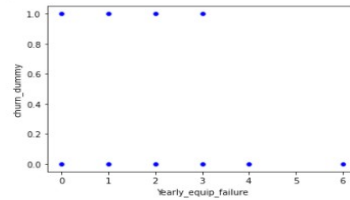
```
[9]: sns.scatterplot(x=df['Age'], y=df['churn_dummy'], color='blue')  
plt.show();
```



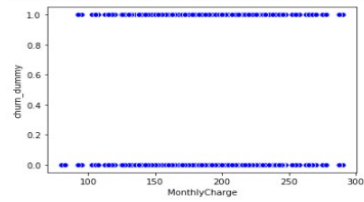
```
[10]: sns.scatterplot(x=df['Outage_sec_perweek'], y=df['churn_dummy'], color='blue')  
plt.show();
```



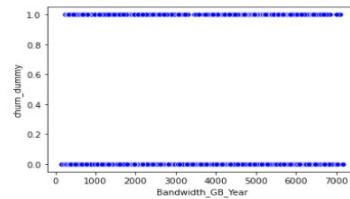
```
[11]: sns.scatterplot(x=df['Yearly equip_failure'], y=df['churn_dummy'], color='blue')  
plt.show();
```



```
[12]: sns.scatterplot(x=df['MonthlyCharge'], y=df['churn_dummy'], color='blue')  
plt.show();
```



```
[13]: sns.scatterplot(x=df['Bandwidth_GB_Year'], y=df['churn_dummy'], color='blue')  
plt.show();
```



Part 4

Initial model:

```
Optimization terminated successfully.
Current function value: 0.357089
Iterations 7

Logit Regression Results
=====
Dep. Variable:    churn_dummy    No. Observations:    10000
Model:            Logit          Df Residuals:         9991
Method:           MLE            Df Model:             8
Date:             Thu, 16 Sep 2021 Pseudo R-squ.:         0.3824
Time:             16:35:36        Log-Likelihood:       -3570.9
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        0.0170     0.014     1.211    0.226    -0.010     0.044
Age             -0.0012     0.001    -0.847    0.397    -0.004     0.002
Income          9.045e-07    1.05e-06     0.858    0.391   -1.16e-06    2.97e-06
Outage_sec_perweek -0.0025     0.010    -0.250    0.803    -0.022     0.017
Yearly equip_failure -0.0418     0.047    -0.887    0.375    -0.134     0.051
MonthlyCharge    0.0342     0.001    38.700    0.000     0.032     0.036
Bandwidth_GB_Year -0.0008    1.97e-05   -41.498    0.000    -0.001    -0.001
gender_dummy     -0.1938     0.059    -3.266    0.001    -0.310    -0.077
Intercept       -4.8619     0.201   -24.229    0.000    -5.255    -4.469
=====
```

Using P-values, we will determine statistical significance and drop any X variables that are not statistically significant. Anything with a recorded P-value less than 0.05 will be dropped which resulted in Children, Age, Income, Outage_sec_perweek, and Yearly equip_failure being removed. We are left with a reduced model which includes 3 X variables: MonthlyCharge, Bandwidth_GB_Year, and gender_dummy. The reduced model is shown below.

Reduced model:

```
Optimization terminated successfully.
Current function value: 0.357282
Iterations 7

Logit Regression Results
=====
Dep. Variable:    churn_dummy    No. Observations:    10000
Model:            Logit          Df Residuals:         9996
Method:           MLE            Df Model:             3
Date:             Thu, 16 Sep 2021 Pseudo R-squ.:         0.3821
Time:             16:35:36        Log-Likelihood:       -3572.8
converged:        True            LL-Null:              -5782.2
Covariance Type:  nonrobust       LLR p-value:          0.000
=====
               coef    std err          z      P>|z|      [0.025    0.975]
-----
MonthlyCharge    0.0342     0.001    38.701    0.000     0.032     0.036
Bandwidth_GB_Year -0.0008    1.97e-05   -41.529    0.000    -0.001    -0.001
gender_dummy     -0.1918     0.059    -3.236    0.001    -0.308    -0.076
Intercept       -4.8958     0.147   -33.228    0.000    -5.185    -4.607
=====
```

The variables chosen were mostly continuous variables because I think they tell a more powerful story and have a more linear relationship with customer churn. It is also more practical to use in a logistic regression analysis since demographic data can bias the report in ways that do

not make sense (this is the reason our data set exuded variables like lat, city, state, etc in the original dataset which are not meaningful in predicting how new customers may churn or not).

Confusion matrix:

```
[18]: matrix_df = pd.read_csv(r'C:\Users\dre2\Desktop\WGU\0208\0208_Task2\0208_task2_data.csv')
      matrix_df = matrix_df[['Children', 'Intercept', 'Age', 'Income', 'Outage_sec_perweek', 'Yearly equip_failure', 'MonthlyCharge', 'Bandwidth_GB_Year', 'gender_dummy', 'churn_dummy']]
      X = matrix_df.iloc[:, 1:-1].values
      y = matrix_df.iloc[:, -1].values

[19]: from sklearn.model_selection import train_test_split
      X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)

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

[20]: LogisticRegression(random_state=0)

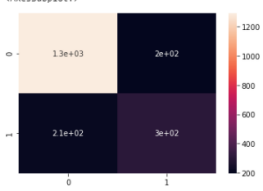
[21]: y_pred = classifier.predict(X_test)

[22]: from sklearn.metrics import confusion_matrix
      matrix = confusion_matrix(y_test, y_pred)
      print(matrix)

[[1289 197]
 [ 210 304]]

[23]: y_predict_test = classifier.predict(X_test)
      new_matrix = confusion_matrix(y_test, y_predict_test)
      sns.heatmap(new_matrix, annot=True)

[23]: <AxesSubplot:~>
```



```
[24]: from sklearn.metrics import classification_report
      print(classification_report(y_test, y_predict_test))
```

	precision	recall	f1-score	support
0	0.86	0.87	0.86	1486
1	0.61	0.59	0.60	514
accuracy			0.80	2000
macro avg	0.73	0.73	0.73	2000
weighted avg	0.79	0.80	0.80	2000

Part 5

Equation for the reduced model:

$$Y = -4.89 + (0.03 * \text{MonthlyCharge}) + (-0.008 * \text{Bandwidth_GB_year}) + (-0.19 * \text{gender_dummy})$$

We see a negative relationship with gender_dummy and Bandwidth_GB_Year, which suggests that females are less likely to churn and that less data used per year will result a customer staying with the company. However, the coefficient is so low for Bandwidth_GB_Year (0.008) that this relationship isn't reliable and I wouldn't consider it meaningful. We see that a higher MonthlyCharge is positively associated with higher churn; a 1 unit increase in MonthlyCharge is associated with a 0.03 increase in churn, since 1 is our 'Yes' response to churn.

The model has a relatively low pseudo R^2 of 0.38 so this would suggest our data is not very good at predicting new values of Churn and our regression line does not fit well. However there is still some relevance to our reduced model since a 0.38 pseudo R^2 isn't too low to consider irrelevant. The limitations of this analysis are mainly contained to the available data set and the variables we have to work with; gaining more information on customer churn data could help. Our reduced model, while having more statistical significance, did not increase R^2 which suggests there still remains some better X variables at explaining the variation in churn.

In conclusion, based on the available data and our reduced model I would suggest that in order to keep customers from churning, the best course of action would be to keep Monthly-Charges as low as possible by providing customers with more pricing flexibility in their use of internet. This is backed up by a very small coefficient for Bandwidth_GB_Year which suggests that customers usage has no effect on churn but how much they pay does.

All references and code are from course lectures + videos, official docs from Python libraries, or from <https://stackoverflow.com/>