­

**Predictive Modeling: Churn Data**

**Task 2: Logistic Regression**

Stephen E. Porter

College of Information Technology, Western Governors University

Dr. William Sewell

March 1, 2022

**Predictive Modeling: Churn Data**

In this paper, I will use a data set containing cleaned customer data from a fictional telecommunications company. The primary purpose is to perform statistical analysis on the cleaned data set to determine which factor (or factors) are the greatest indicator of customer turnover.

# Part I: Research Question

## A1. Research Question

Which factor (or factors) have the greatest effect on customer churn? Acquiring new customers is more costly than keeping current ones. (AltlexSoft, 2020) If customers discontinue their service out of low satisfaction, the negative opinions they would share with friends and family will serve as a deterrent. (Qualtrics 2022)

## A2. Objectives & Goals

The objective of this analysis is to use multiple logistic regression to create a model that can accurately predict customer churn. By providing this tool, leadership will be able to take a more targeted approach to addressing customer concerns to reduce customer churn.

# Part II: Method Justification

## B1. Summary of Assumptions

For multiple logistic regression, the following assumptions are made:

1. The target variable is binary
2. The observations are independent of each other
3. No multicollinearity among explanatory variables
4. There are no extreme outliers
5. There is a linear relationship between the each explanatory variable and the logit of the target variable
6. The sample size is large enough to draw valid conclusions

(Bobbit, 2020)

## B2. Tool Benefits

R (and the RStudio IDE) is a powerful tool for statistical analysis, and the language makes for an excellent choice for logistic regression. Several libraries chosen are tidyverse, broom, ggplot2, fastDummies, caret, car, corrplot, Hmisc, Metrics, and cowplot.

The core functions that R provides are *glm* and *predict*. The function *glm* is used to fit logistic models and can handle multiple predictor variables easily. Once a model is obtained from the *glm* function applied to a training data set, *predict* can be used on a testing data set to determine how well the model fits the data. The *glm* function only accepts numerical predictor variables, but categorical variables can still influence the target variable. To handle categorical data, the fastDummies package has the *dummy­\_cols* function that quickly converts these columns to numerical values. These features make using R to perform multiple linear regression simple.

## B3. Appropriate Technique

They target variable, Churn, is essentially a binary variable (with Yes corresponding to 1 and No to 0). For this main reason, logistic regression is more appropriate than linear regression. Multiple logistic regression will allow me to determine which of these variables are worth including in the model to ensure it is valid without overfitting to this particular data set. Variance Inflation Factor and correlation matrices are used to reduce multicollinearity, corrplots and scatterplots are used to visualize the data, and a confusion matrix helps determine accuracy of the final model.

# Part III: Data Preparation

## C1. Data Goals

To answer the research question, the data must be tidied by removing irrelevant columns (such as ID numbers) and columns with too many unique entries to create dummy variables (such as City and County). Dummy variables will be created for the remaining categorical variables (such as Gender, Contract, and InternetService) and the new column headers will need to be tidied as well so they can be selected by name as part of the linear regression process. The data set will then be split into a training set and a testing set (70/30 split), respectively.

Once the data set has been tidied and split, it is ready for regression analysis.

## C2. Summary Statistics

Once the dummy columns have been introduced to the data frame, there are 118 columns – far too many to list reasonably and repeatedly. At the end of the process, the ten (10) variables that were determined to be worth retaining for the model are:

* Bandwidth\_GB\_Year
* State\_RI
* Gender\_Male
* Techie\_No
* Contract\_Month\_to\_month
* InternetService\_DSL
* OnlineBackup\_No
* DeviceProtection\_No
* TechSupport\_No
* PaymentMethod\_Electronic\_Check

Below is the summary statistics output for all 118 variables.

Text, table

Description automatically generated Table, calendar

Description automatically generated Table

Description automatically generated Table

Description automatically generated Text

Description automatically generated

## C3. Steps to Prepare the Data

To prepare the data, data frame will be reviewed for nulls and the column names will be renamed as needed. Columns named Item1 through Item8 will be given more meaningful names. Other columns will be dropped as they will not provide useful information (such as CaseOrder and the various ID codes) or creating dummy variables to model the data in the columns would be cause for high multicollinearity (such as City, and County). Dummy variables will need to be created for categorical data, and those columns will need to be renamed to remove characters that would interfere with the code running successfully (like spaces, hyphens, and parenthesis). The data set will then be ready to be split into testing and training sets containing 70% and 30% of the data, respectively.

Once the data training and testing sets are ready, the initial logistic model will be created from all predictor variables and reduced to a more manageable collection.

Please see included code output file with the submission for more detail.

## C4. Visualizations

The data visualizations below show histograms of Bandwidth\_GB\_Year and Contacts values followed by logistic models fit to MonthlyCharge vs. Churn\_Yes and Tenure vs. Churn\_Yes. Graphical user interface, chart

Description automatically generated

Figure 1. Univariate and Bivariate Data Visualizations

## C5. Prepared Data Set

Please see the files titled *D208\_dfTrain.csv* and *D208\_dfTest.csv* included in the submission for the prepared data sets.

# Part IV: Model & Comparison Analysis

## D1. Initial Model

In my code, I refer to the final version of the multiple linear regression model as “*reducedModel*” and all iterations of the model prior to the final as “*initialModel*,” “*initialModel2*”, and so on (I realize the inaccuracy of the name *initialModel2*; however, it allowed for consistency in naming throughout the process). With so many predictor variables, there was no confidence in the p-values correctly identifying which are significant. Therefore, the predictor variables needed to be analyzed and systematically reduced.

The formula for first version of the initial linear regression model (*initialModel*) was created by taking the column names of the normalized data frame (*dfTrainNorm*) using a combination of the **colnames** and **paste** functions as there were too many to list manually.

The *initialModel* has an R-squared value of 1 which implies it is over-fitted to this specific data set. The summary graphs for *initialModel* (Figure 2) confirm that the model is in fact a poor fit for the data.

Graphical user interface

Description automatically generated

Figure 2. Summary graphs of *initialModel*

## D2. Justification of Model Reduction

The *initialModel* was reduced systematically using multiple tools to view multicollinearity and correlation. The *initialModel* contained too many variables for the Variance Inflation Factor (VIF) function to run properly. I created a data frame from *initialModel* and removed predictor variables that were listed as “NA” meaning these were highly correlated to another (these tended to be the “Yes” versions of the dummy variables, which is a logical result).

I then applied to the same steps to create a new formula for the linear regression model which I called *initialModel2*. This version allowed VIF to run which confirmed that there was still high multicollinearity. To gain a better understanding of how the variables were correlated, I attempted to view a correlation matrix and **corrplot** of the model. However, were too many predictor variables for the matrix to be reviewed by hand or for image to be decipherable. I then found a custom function from the Statistical Tools for High-Throughput Data Analysis website called **flattenCorrMatrix** (Kassambara, 2018) that turns the correlation matrix into a data frame that can then be manipulated more easily. I then filtered the data frame for correlation values above 0.75 or below -0.75 and reviewed these values manually. This showed that there was a high correlation between Tenure and Bandwidth\_GB\_Year, Gender\_Female and Gender\_Male, and Churn\_No and Churn\_Yes. I chose to remove the first option from each pair: Tenure, Gender\_Female, and Churn\_No (note: Churn\_No was always planned to be dropped because it is redundant with Churn\_Yes).

After removing these columns, I repeated my process to create *initialModel3*. After reviewing the VIF scores for this version of the model, I created a data frame of the values and filtered for predictor variable names that had VIF scores above 5 (Bobbitt, 2019). I removed those predictor variables and repeated the process a fourth time to obtain *initialModel4*. There were no high correlation values or VIF scores for this model which means the p-values should be correctly identifying significant predictor variables.

## D3. Reduced Multiple Regression Mode

To arrive at the reduced model, I selected the predictor variables from *initialModel4* that had a significant p-value. The *reducedModel* contains fourteen predictor variables that were shown to be significant after removing multicollinearity and correlation (are listed in C2. Summary Statistics).

Graphical user interface

Description automatically generated

Figure 3. Summary graphs of *initialModel4*

Graphical user interface, application

Description automatically generated

Figure 4. Summary graphs of *reducedModel*

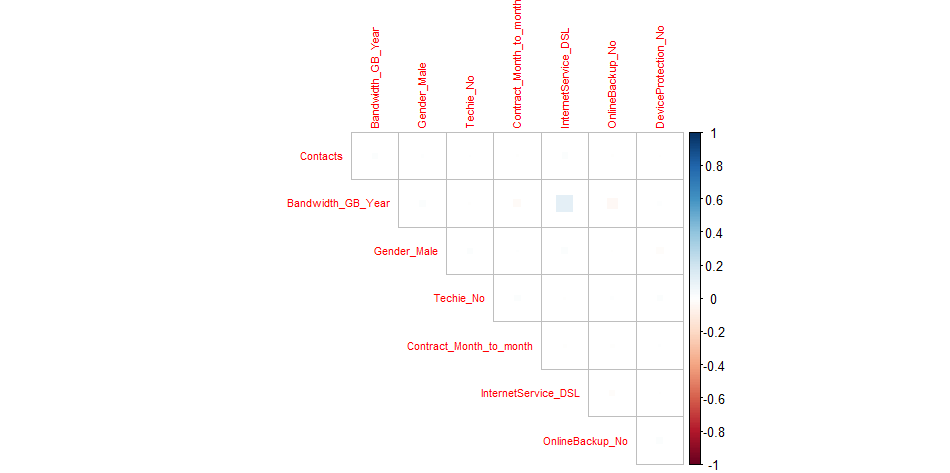


Figure 5. Visualization of Correlation Matrix of *reducedModel*

Graphical user interface

Description automatically generated with low confidence

Figure 6. Plots of Predictor Variables vs Residuals of *reducedModel*

## E1. Model Comparison

By comparing the summary graphs of *initialModel* (Figure 1), *initialModel4* (Figure 3), and *reducedModel* (Figure 4) it is clear that removing unnecessary predictor variables has led to a better model. There is a marginal difference between *initialModel4* and *reducedModel* graphs which highlights there were still unnecessary predictor variables to remove; however, the difference was not so significant to be concerned the *reducedModel* does not fit the data.

## E2. Output and Calculations

The confusion matrix and accompanying statistics show a model accuracy of 80.67%, sensitivity of 89.02%, and specificity of 57.48% which imply a good fit. See the attached code for all calculations.

## E3. Code

Please see the file included in my submission for the complete code.

# Part V: Data Summary and Implications

## F1. Results

The process reduced the total number of predictor variables from over 100 down to a more manageable ten (10). Running the **predict** function with the *reducedModel* and testing data set (*dfTest*) resulted in a model with a strong fit to the data. This model can be used to predict the tenure of users based on the ten significant predictor variables.

## F2. Recommendations

This model could be used to identify customers at risk of churning. If the model predicts they have churned, but they in fact have not yet, these customers would be prime targets to contact by customer service. Customer service *should* reach out to these users to determine satisfaction and/or to offer incentives to retain their service. With over an 80% accuracy rate, leadership should feel confident time spent contacting these customers would be a worthwhile use of resources to retain these customers.

# Part VI: Demonstration

## G. Panapto Demonstration

Please view video included in the submission.

## H. Sources of Third-Party Code

Kassambara, A. (2018). Correlation matrix : A quick start guide to analyze, format and visualize a correlation matrix using R software. STHDA. Retrieved February 12, 2022, from <http://www.sthda.com/english/wiki/correlation-matrix-a-quick-start-guide-to-analyze-format-and-visualize-a-correlation-matrix-using-r-software>

## I. Sources

**References**

Bobbitt, Z. (2020, October 13). The 6 assumptions of logistic regression (with examples). Statology. Retrieved March 1, 2022, from <https://www.statology.org/assumptions-of-logistic-regression/>

AltexSoft. (2020, February 27). Customer churn prediction for subscription businesses using Machine Learning: Main Approaches and Models. AltexSoft. Retrieved November 11, 2021, from <https://www.altexsoft.com/blog/business/customer-churn-prediction-for-subscription-businesses-using-machine-learning-main-approaches-and-models/>

Kassambara, A. (2017, September 1). GGPLOT2 - easy way to mix multiple graphs on the same page. STHDA. Retrieved October 28, 2021, from <http://www.sthda.com/english/articles/24-ggpubr-publication-ready-plots/81-ggplot2-easy-way-to-mix-multiple-graphs-on-the-same-page/>

Kassambara, A. (2018). Correlation matrix : A quick start guide to analyze, format and visualize a correlation matrix using R software. STHDA. Retrieved February 12, 2022, from <http://www.sthda.com/english/wiki/correlation-matrix-a-quick-start-guide-to-analyze-format-and-visualize-a-correlation-matrix-using-r-software>

rdata.lu Blog | Data science with R. (2018, January 4). Churn analysis – part 1: Model selection: R-bloggers. R-Bloggers. Retrieved February 12, 2022, from <https://www.r-bloggers.com/2018/01/churn-analysis-part-1-model-selection/>

What is customer churn? learn to measure &amp; prevent it. Qualtrics. (2022, February 22). Retrieved March 1, 2022, from <https://www.qualtrics.com/experience-management/customer/customer-churn/>