­

**Predictive Modeling: Churn Data**

Stephen E. Porter

College of Information Technology, Western Governors University

Dr. William Sewell

February 23, 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 tenure? While an analysis of tenure will not directly correlate to customer churn, it may provide valuable insights into what customers are likely to stay with this service for a greater length of time. Acquiring new customers is more costly than keeping current ones. (AltlexSoft, 2020)

## A2. Objectives & Goals

The primary goal of this analysis is to use multiple linear regression to determine a model that will accurately predict customer tenure. This will enable company leadership to take more targeted action to address issues around customer tenure.

# Part II: Method Justification

## B1. Summary of Assumptions

The following assumptions are made as part of multiple linear regression:

1. there is a linear relationship between independent and dependent variables,
2. there is no multi-collinearity,
3. the observations are independent,
4. the residuals have constant variance, and
5. the residuals are normally distributed.

(Bobbitt, 2021). Some of these can be addressed as part of the data preparation to ensure the assumptions are met.

## B2. Tool Benefits

My analysis will be performed using R and its relevant libraries. R (and the RStudio IDE) is a powerful tool for statistical analysis. Libraries used are tidyverse, broom, ggplot2, fastDummies, caret, car, corrplot, Hmisc, Metrics, and cowplot.

The core functions that R provides are *lm* and *predict*. The function *lm* is used to fit linear models and can handle multiple predictor variables easily. Once a linear model is obtained from the *lm* 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 *lm* 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

The target variable, Tenure, is continuous and can be influenced by many predictor variables in the data set. Multiple linear regression will allow me to determine which of these variables are strong predictors and generate a model to make predictions. Variance Inflation Factor and correlation matrices are used to reduce multicollinearity; corrplots and scatterplots are used to visualize the data; R-squared and root mean squared error (RMSE) are used to determine fit – all of which are statistical tools to confirm the final model is appropriate.

# 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) and normalized to ensure large values (like Income) do not obscure the comparatively smaller values (like number of Children).

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

## C2. Summary Statistics

The target variable for this model is Tenure and this will be compared against all remaining variables. Because dummy variables were created for categorical variables, the initial model has too many variables (66) to list in a reasonable manner, so instead the variables identified for the reduced model are provided:

1. Children
2. Age
3. Bandwidth\_GB\_Year
4. Gender\_Female (dummy variable for Gender)
5. Churn\_No (dummy variable for Churn)
6. Techie\_No (dummy variable for Techie)
7. Contract\_Month\_to\_month (dummy variable for Contract)
8. Tablet\_No (dummy variable for Tablet)
9. InternetService\_DSL (dummy variable for InternetService)
10. Multiple\_No (dummy variable for Multiple)
11. OnlineSecurity\_No (dummy variable for OnlineSecurity)
12. OnlineBackup\_No (dummy variable for OnlineBackup)
13. DeviceProtection\_No (dummy variable for DeviceProtection)
14. PaperlessBilling\_No (dummy variable for PaperlessBilling)

This list was created through repeated review of multicollinearity and removal of those variables to arrive at these predictor variables. The final version (called *reducedModel*) has an R-squared value of 0.9961 and root mean square error (RMSE) value of approximately 0.0241 which means the model fits the data well.

The summary statistics for each variable are provided below.

A screenshot of a computer

Description automatically generated with low confidenceA screenshot of a computer

Description automatically generated with low confidence`Table

Description automatically generatedTable

Description automatically generatedTable

Description automatically generatedTable

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated with medium confidence

## 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, State, 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, and normalized (ensuring Assumption 5 is true).

Once the data sets are ready, the initial linear model will be created from all predictor variables and reduced to a more manageable collection (ensuring Assumption 2 is true).

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

## C4. Visualizations

Histograms of Children (discrete variable), Income (continuous variable), and Tenure (target variable) as well as scatterplots of Children vs. Tenure, Income vs. Tenure, and Bandwidth\_GB\_Year vs. Tenure are provided below.

Graphical user interface

Description automatically generated

## C5. Prepared Data Set

Please see the files titled *D208\_dfTrainNorm.csv* and *D208\_dfTestNorm.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 1) confirm that the model is in fact a poor fit for the data.

A screenshot of a computer

Description automatically generated with medium confidence

Figure 1. Summary of *initialModel*

Table

Description automatically generatedTable

Description automatically generatedTable

Description automatically generatedTable

Description automatically generatedTable

Description automatically generatedTable

Description automatically generated

## 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 and created *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 which I chose to leave as-is because Tenure is the target variable. The other pair that was still highly correlated was Gender\_Female and Gender\_Male, and I chose to remove the latter.

After removing Gender\_Male, 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 Model

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). This model has an R-squared value of 0.9961 and RMSE of approximately 0.0241 implying a good fit.

Graphical user interface

Description automatically generated

Figure 2. Summary of *initialModel4*

Graphical user interface

Description automatically generated

Figure 3. Summary of *reducedModel*

Chart, bar chart

Description automatically generated

Figure 4. **corrplot** of *reducedModel* correlation matrix

Graphical user interface, application

Description automatically generated

Figure 5. 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

This model has an R-squared value of 0.9961 and RMSE of approximately 0.0241 implying a good fit. The code (see attached) contains 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 66 down to a more manageable 14. Running the **predict** function with the *reducedModel* and testing data set (*dfTestNorm*) 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 fourteen significant predictor variables – the strongest of which appears to be Bandwidth\_GB\_Year.

## F2. Recommendations

This model could be used to identify current users whose actual tenure does not match the predicted tenure and/or predict the tenure of new customers. If a user’s actual tenure is low, but the model predicts it will be high this is a user who is unlikely to churn. The company likely does not need to take any action with these users but can have customer service reach out to ensure they are still satisfied with their service. If the user’s actual tenure is higher than their predicted tenure, this user may be at risk for churn. Customer service *should* reach out to these users to determine satisfaction and/or to offer incentives to retain their service. Predicting the tenure of new customers would follow similar logic; the company would have an estimate of when the customer may discontinue their service and can potentially intervene *before* the customer has made that decision.

# 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. (2021, November 16). The five assumptions of multiple linear regression. Statology. Retrieved January 10, 2022, from <https://www.statology.org/multiple-linear-regression-assumptions/>

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/>

Sharma, A. (2021, May 2). Predictive analysis using multiple linear regression. Medium. Retrieved February 12, 2022, from <https://medium.com/data-science-on-customer-churn-data/predictive-analysis-using-multiple-linear-regression-b6b3b79b36b6>