Group Project

(Signature Assignment Report)

Group Alpha:

Aytaj Khankishiyeva

Fatima Nurmakhamadova

David Belyaev

Vaibhav Arora

Xiaolu Shen

Prof. Valeriy Shevchenko

ALY 6015 - 21495 Intermediate Analytics

College of Professional Study, Northeastern University

February 17, 2022

Shen.Xiao@northeastern.edu

Introduction

The main goal of this report is to build a predictive model to predict the churn rate of the telecom company. To achieve this, we have built and compared 2 supervised machine learning models with two methods that identify the critical factors affecting the churn rate. The question we are answering in this report is “What are the significant predictor variables that affect the Telco customers' churn rate ?”

Methods

The Logistic Regression and LASSO Regularization Regression models are used and then compared using the AUC values to find the better performing machine learning model. We are using a Logistic regression model because our response variable is categorical and binomial. We are using the LASSO regression model to prevent overfitting and make the model flexible by shrinking the coefficients to zero. Although the Lasso is less flexible but more interpretable while the GLM is balanced between flexibility and interpretability.

Data Set

The data set used is <Telco Customer Churn> that was retrieved from Kaggle, and has 7043 observations of 21 variables. The dataset contains information about Telecom customers, their gender, tenure, charges, payment method, usage of internet, streaming service, and churn rate along with other variables. All variables are categorical except monthly and total charges, and tenure, which is numerical.

Data Cleaning and Preparations

First, the missing values in the data set should be checked. The following R screenshot shows using <is.na> and summary function to check and display whether there are and where the missing exist:

```
> any(is.na(OGdata))
[1] TRUE
> #Checking the type of the variables and lengths
> summary(OGdata) # 11 Na values exist in the column 'TotalCharges'
   customerID      gender      SeniorCitizen    Partner     Dependents      tenure      PhoneService         MultipleLines     InternetService           OnlineSecurity           OnlineBackup           DeviceProtection
 0002-ORFBO:   1   Female:3488   Min.   :0.0000   No :3641   No :4933   Min.   : 0.00   No : 682       No              :3390   DSL        :2421   No              :3498   No              :3088   No              :3095
 0003-MKNFE:   1   Male  :3555   1st Qu.:0.0000   Yes:3402   Yes:2110   1st Qu.: 9.00   Yes:6361       No phone service: 682   Fiber optic:3096   No internet service:1526   No internet service:1526   No internet service:1526
 0004-TLHLJ:   1                 Median :0.0000                         Median :29.00                  Yes             :2971   No         :1526   Yes             :2019   Yes             :2429   Yes             :2422
 0011-IGKFF:   1                 Mean   :0.1621                         Mean   :32.37
 0013-EXCHZ:   1                 3rd Qu.:0.0000                         3rd Qu.:55.00
 0013-MHZWF:   1                 Max.   :1.0000                         Max.   :72.00
 (Other)   :7037
       TechSupport            StreamingTV            StreamingMovies               Contract    PaperlessBilling                PaymentMethod  MonthlyCharges   TotalCharges    Churn
 No              :3473   No              :2810   No              :2785   Month-to-month:3875   No :2872        Bank transfer (automatic):1544   Min.   : 18.25   Min.   :  18.8   No :5174
 No internet service:1526   No internet service:1526   No internet service:1526   One year      :1473   Yes:4171        Credit card (automatic) :1522   1st Qu.: 35.50   1st Qu.: 401.4   Yes:1869
 Yes             :2044   Yes             :2707   Yes             :2732   Two year      :1695                   Electronic check        :2365   Median : 70.35   Median :1397.5
                                                                                                               Mailed check            :1612   Mean   : 64.76   Mean   :2283.3
                                                                                                                                               3rd Qu.: 89.85   3rd Qu.:3794.7
                                                                                                                                               Max.   :118.75   Max.   :8684.8
                                                                                                                                                                NA's   :11
>
```

As a result, the original data set has 11 missing values in the column <TotalCharges>. Considering the total number of observations in the variable, substituting those missing values with the column means will not have any statistically significant impact on the distribution of the variable. Therefore, missing values will be substituted with the column mean.

```
> # Substituting missing values with the column mean (for all the columns)
> for (cols in colnames(OGdata)) {
+   if (cols %in% names(OGdata[,sapply(OGdata, is.numeric)])) {
+     OGdata<-OGdata%>%
+       mutate(!!cols := replace(!!rlang::sym(cols), is.na(!!rlang::sym(cols)),
+                                mean(!!rlang::sym(cols), na.rm=TRUE)))
+   }
+ }
>
```

As the substitution was conducted, recheck the situation of missing values:

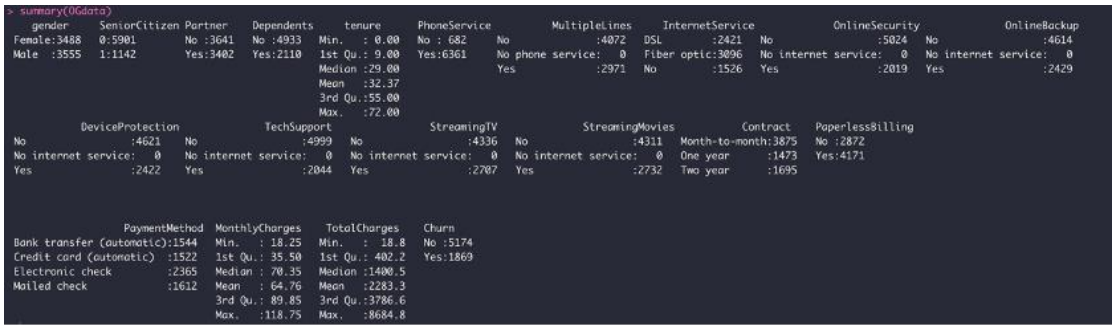
```
+ }
> # Checking for missing values to make sure no more missing values are present
> any(is.na(OGdata))
[1] FALSE
> # target response: Churn
> table(OGdata$Churn) # 5174 No, 1869 Yes

  No  Yes
5174 1869
>
```

We also check the response variable distribution.

Second, cut the no-use columns. Because the first column <customerID> only contains the unique identifiers for customers, which is no use for the further EDA and modeling, as a result, remove the column. Now, the data set has 20 variables with 7043 records.

From the summary function, we observe that there are three levels in some of the variables: internetService, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies, Contract. However, according to the glossary for the data set, there are only two levels for these columns, “Yes” and “No”. Thus, we take this as mistakenly recorded data, and convert the third level “No service” and other negative statements into "No".

Exploratory Data Analysis



The dataset summary is presented on the screenshot above. As can be seen, there are a total of 20 variables with 7,043 observations for each one of them. Almost

all variables (17 out of 20) are categorical. The remaining variables are discrete and continuous. The summary for discrete and continuous variables (<Tenure>, <Monthly Charges>, and <Total Charges>) is presented on the screenshot below.

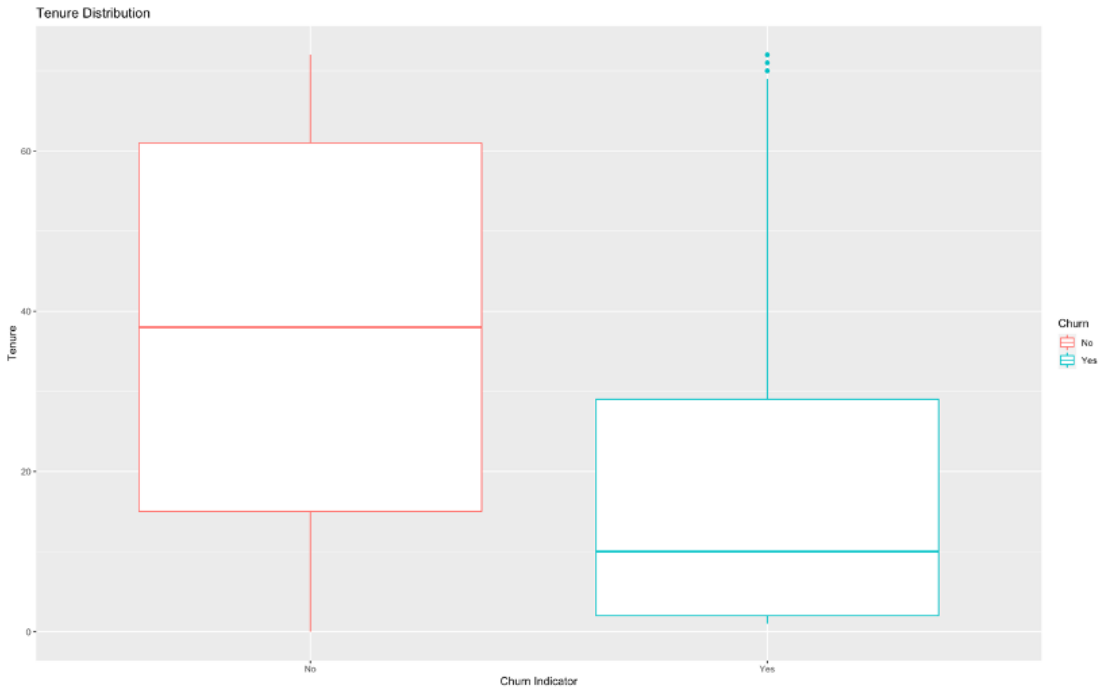
```
> describe(OGdata$tenure)
   vars    n  mean    sd median trimmed   mad min max range skew kurtosis   se
X1    1 7043 32.37 24.56     29   31.43 32.62   0  72    72 0.24    -1.39 0.29
> describe(OGdata$MonthlyCharges)
   vars    n  mean    sd median trimmed   mad   min    max range  skew kurtosis   se
X1    1 7043 64.76 30.09  70.35   64.97 35.66 18.25 118.75 100.5 -0.22    -1.26 0.36
> describe(OGdata$TotalCharges)
   vars    n   mean   sd  median trimmed    mad  min    max range skew kurtosis    se
X1    1 7043 2283.3 2265 1400.55 1970.38 1812.4 18.8 8684.8  8666 0.96    -0.23 26.99
```

As can be seen from the screenshot above, the statistical summaries are presented for each of the variables. There are a total of 7,043 observations for each of the variables, with means 32.37 for <Tenure>, 64.76 for <Monthly Charges>, and 2,283.3 for <Total Charges>. The standard deviations are 24.56, 30.09, and 2,265, and medians are 29, 70.35, 1,400.55 respectively. A relative to the distribution of the large value of the standard deviation for <Total Charges> indicates that the data for this variable is very dispersed.

To put the statistical findings into actual business insights, customers stay with the company on average for 32 months out of a maximum of 72 months. Considering the skewness and the kurtosis of the variable <Tenure>, we can conclude that most customers stay with the company on average for 32 months plus-minus 24 months.

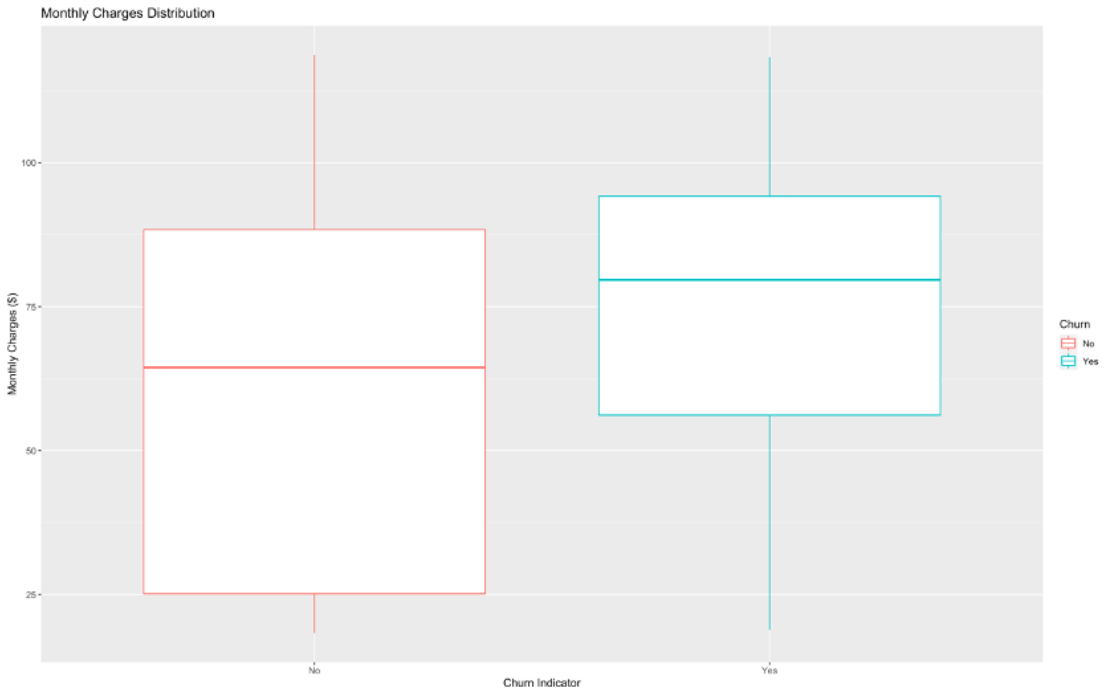
Box Plots and Histograms

The variables that were discussed will be visualized in a form of a boxplot, which is a perfect visualization for understanding the distribution of the variable. All of the boxplots are plotted against the target variable <Churn> to understand if certain trends are present or not.

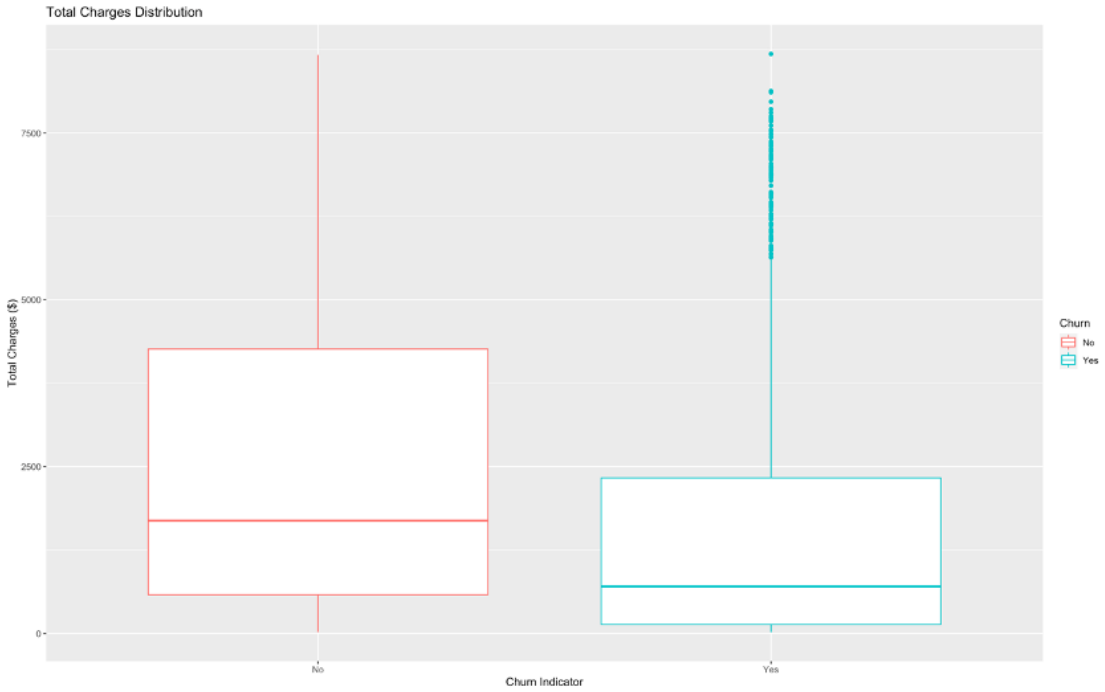


The boxplot presented in the figure above represents the <Tenure> distribution plotted against the target variable <Churn>. From the visualization of those two variables, the relationship is clear. Customers tend to churn in their first 2 years (24 months) with some outliers that stay with the company for up to 70 months and more.

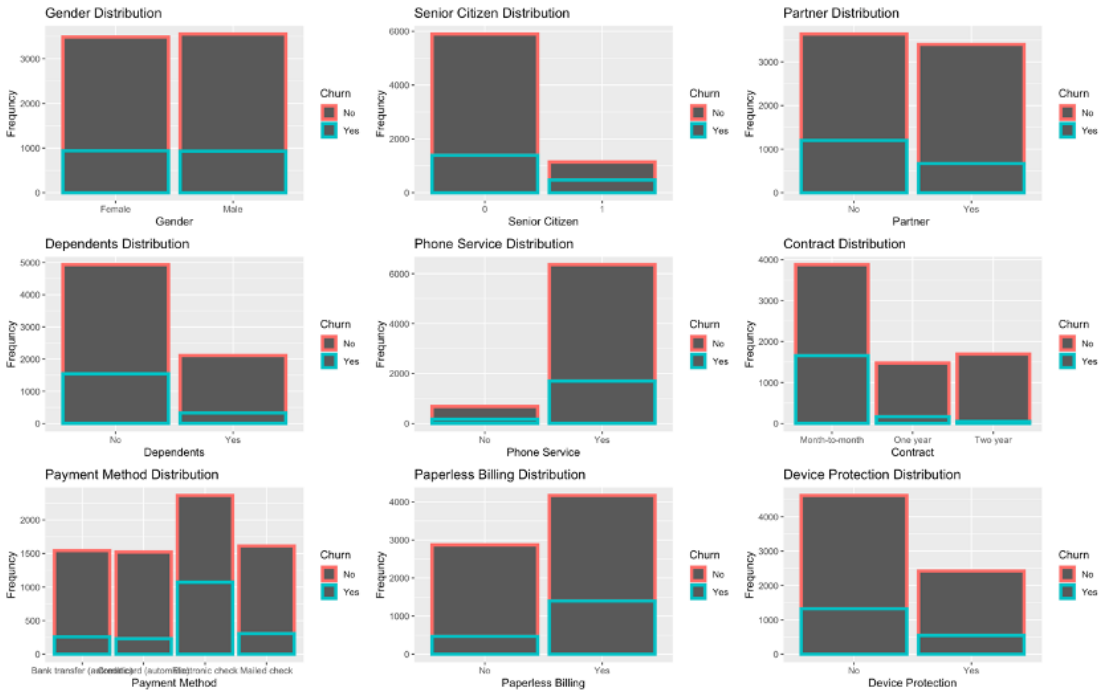
The boxplot presented in the figure below represents the <Monthly Charges> distribution plotted against the target variable <Churn>. From the visualization of those two variables, the relationship is clear. Customers with the higher monthly charges tend to churn rather than customers with the lower monthly charges.



The final boxplot presented in the figure below represents the <Total Charges> distribution plotted against the target variable <Churn>. From the visualization of those two variables, the relationship is not as clear as in the previous case with <Monthly Charges>. There is no clear relationship between those two variables identified. That might be explained by the human factor, where people tend to notice monthly spending, however, if they are accumulated through the whole year, the decision to churn or not is not affected so much.



The following bar plots show the distributions of categorical variables against churn indicators.



A total of nine categorical variables were included in this set of bar plots. Those variables are <Gender>, <SeniorCitizen>, <Partner>, <Dependents>, <PhoneService>, <Contract>, <PaymentMethod>, <PaperlessBilling>, and <DeviceProtection>. Overall, most of the variables have a certain relationship with the variable <Churn>, however, variables like <Gender>, for example, clearly have no relationship with <Churn>. In the dataset, the male category is a bit larger than the female, however, the number of <Churns> is identical in both groups. The variable <PaymentMenthod> contains one category (Electronic Check) that has a higher effect on <Churn> compared to the rest of the categories. Apart from the variables mentioned in this part, the remaining seven variables that are presented on the set of plots all have a certain relationship. Apart from the relationship between the variables, the variables category breakdown is good to look at. For example, gender is split into two categories (male and female) equally, however, the variable <PhoneService> is different. The categories are ‘Yes’ and ‘No’, and the category ‘Yes’ is much more dominant in the variables.

Pair Plot

The plot presented above contains a summary of visualized information in the form of the matrix for a total of six variables. There are several key takeaways from this plot. Firstly, the total charges highly depend on the presence or absence of the partner. Secondly, dependents have a relationship with tenure. The presence of dependents leads to a higher tenure. Similar information can be extracted from this plot for all other variables used in it.

Predictive Models

This part focuses on applying the predictive models to realize the predictions on customers' churn situation and interpreting the results of modeling. We have used 2 predictive models: Logistic Regression and LASSO Regularization Regression. The variable <Tenure> was decided to be grouped by years, instead of months. The purpose of this modification is to eliminate the creation of 71 dummy variables.

Data Preparations

The main goal of the predictive models is to classify the customer's churn indicators (would churn or would not churn). Here we first convert the response variable <Churn> into factors to meet the classification models' input requirements:

```
# convert the response type into factors
OGdata$Churn <- as.factor(OGdata$Churn)
```

In addition, for the variable <tenure>, the original data contains these as discrete values representing how many months has the customer been with the company. In the following report, we would use original values (month scale) in the

section of exploratory data analysis, however, for not creating too many dummy variables in the section of predictive models, we convert tenure values into factors <ten_fact> and change them at the scale of years:

```
################################################################################
# Variable 'Tenure' modifications
################################################################################

# splitting tenure into factors and mutating the column
OGdata <- mutate(OGdata, ten_fact = tenure)

#mutating tenure column to 0-1 years of tenure
OGdata$ten_fact[OGdata$ten_fact >=0 & OGdata$ten_fact <= 12] <- '0-1year'
#mutating tenure column to 1-2 years of tenure
OGdata$ten_fact[OGdata$ten_fact> 12 & OGdata$ten_fact <= 24] <- '1-2years'
#mutating tenure column to 2-3 years of tenure
OGdata$ten_fact[OGdata$ten_fact > 24 & OGdata$ten_fact <= 36] <- '2-3years'
#mutating tenure column to 3-4 years of tenure
OGdata$ten_fact[OGdata$ten_fact > 36 & OGdata$ten_fact <= 48] <- '3-4years'
#mutating tenure column to 4-5 years of tenure
OGdata$ten_fact[OGdata$ten_fact > 48 & OGdata$ten_fact <= 60] <- '4-5years'
#mutating tenure column to 5-6 years of tenure
OGdata$ten_fact[OGdata$ten_fact > 60 & OGdata$ten_fact <= 72] <- '5-6years'

#converting tenure into factors
OGdata$ten_fact <- as.factor(OGdata$ten_fact)
```

After setting the random seeds for getting the same outputs when we re-run the models, here we divide the processed data into a test set and a training set at the ratio of 80/20 to compare and test the predictive models' accuracy and whether it has the problem of overfitting.

```
# Get 80% of random row numbers
dt <- sample(nrow(OGdata), nrow(OGdata) * 0.8)

# Get training data which include 80% of rows
train <- OGdata[dt,]
# Get rest 20% test data
test <- OGdata[-dt,]
```

```
> # Number of rows in original dataset
> dim(OGdata)
[1] 7043   20
> # Verify number of rows in training data set
> dim(train)
[1] 5634   20
> # Verify number of rows in testing data set
> dim(test)
[1] 1409   20
```

Now, we have 1409 records in the test set and 5634 records in the training set. Also, as we would use logistic regression and LASSO regression model to find the different lambda values, here we create input matrices for the <glmnet> and <cv.glmnet> functions, and pass the response values (<Churn>) separately into <trainY> and <testY>:

```r
# use model.matrix to seperately build matrixes for training and test
trainX <- model.matrix(Churn ~., train)

# somehow the matrix has an additional column <intercept>, remove it
trainX <- trainX[,-1]

# similar as the previous step, create input matrix for input matrix of test set
testX <- model.matrix(Churn ~., test)
testX <- testX[,-1]


# pass the response <Grad.Rate> values into the train and test matrices:
trainY <- train$Churn
testY <- test$Churn
```

GLM: Logistic Regression

The first model we applied is logistic regression using the generalized linear model function, to build an effective model that can predict customer churn. The variable <Tenure> was decided to be grouped by years, instead of months. The purpose of this modification is to eliminate the creation of 71 dummy variables. We used the same split test and train data. First, we created the full regression model fitting with all 19 variables to find out the most significant ones where our dependent variable is the 'Churn'. Then, based on the results of this model, we fitted the reduced model with only 6 significant variables.

```r
#- Run logistic regression using binomial family and logit link function
# 1- Fit a logistic regression model with all variables
fit.full <- glm(Churn ~ ., data = train, family=binomial(link="logit"))

#Show fitted model
summary(fit.full)


# 2- Fit a logistic regression model with only significant variables
fit.reduced <- glm(Churn ~ SeniorCitizen + MultipleLines+Contract+
                     PaperlessBilling+PaymentMethod+ten_fact,
                   data = train, family=binomial(link="logit"))

#Show fitted model
summary(fit.reduced)
```

The AIC results of the full model showed lower AIC 4737.1 than the reduced model AIC 4985.6. So, the full model with all predictors seems to be a better model

than the reduced model with only significant predictors.

```
> #Show fitted model
> summary(fit.full)

Call:
glm(formula = Churn ~ ., family = binomial(link = "logit"), data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0414  -0.6719  -0.2902   0.6618   3.0594

Coefficients:
                                        Estimate Std. Error z value Pr(>|z|)
(Intercept)                             3.808e-01  9.084e-01   0.419 0.675056
genderMale                             -1.268e-02  7.268e-02  -0.174 0.861512
SeniorCitizen1                          2.632e-01  9.403e-02   2.800 0.005115 **
PartnerYes                             -3.921e-02  8.666e-02  -0.452 0.650920
DependentsYes                          -1.276e-01  1.005e-01  -1.269 0.204332
PhoneServiceYes                         1.422e-01  7.280e-01   0.195 0.845150
MultipleLinesYes                        4.397e-01  1.997e-01   2.202 0.027656 *
InternetServiceFiber optic              1.655e+00  8.959e-01   1.848 0.064650 .
InternetServiceNo                      -1.599e+00  9.055e-01  -1.766 0.077338 .
OnlineSecurityYes                      -2.128e-01  2.009e-01  -1.059 0.289434
OnlineBackupYes                         7.942e-02  1.972e-01   0.403 0.687083
DeviceProtectionYes                     1.909e-01  1.985e-01   0.962 0.336253
TechSupportYes                         -1.325e-01  2.010e-01  -0.659 0.509924
StreamingTVYes                          5.576e-01  3.656e-01   1.525 0.127229
StreamingMoviesYes                      5.821e-01  3.675e-01   1.584 0.113232
ContractOne year                       -7.538e-01  1.200e-01  -6.284 3.29e-10 ***
ContractTwo year                       -1.638e+00  2.015e-01  -8.129 4.34e-16 ***
PaperlessBillingYes                     3.218e-01  8.353e-02   3.853 0.000117 ***
PaymentMethodCredit card (automatic)   -4.622e-02  1.261e-01  -0.366 0.714074
PaymentMethodElectronic check           3.544e-01  1.058e-01   3.349 0.000811 ***
PaymentMethodMailed check              -6.594e-03  1.282e-01  -0.051 0.958993
MonthlyCharges                         -2.843e-02  3.563e-02  -0.798 0.424902
TotalCharges                           -1.104e-04  6.827e-05  -1.618 0.105738
ten_fact1-2years                       -8.184e-01  1.173e-01  -6.980 2.95e-12 ***
ten_fact2-3years                       -1.191e+00  1.669e-01  -7.136 9.58e-13 ***
ten_fact3-4years                       -9.287e-01  2.221e-01  -4.181 2.90e-05 ***
ten_fact4-5years                       -1.138e+00  2.876e-01  -3.955 7.65e-05 ***
ten_fact5-6years                       -1.272e+00  3.809e-01  -3.339 0.000841 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 6525.6  on 5633  degrees of freedom
Residual deviance: 4681.1  on 5606  degrees of freedom
AIC: 4737.1

Number of Fisher Scoring iterations: 6
```

```
> #Show fitted model
> summary(fit.reduced)

Call:
glm(formula = Churn ~ SeniorCitizen + MultipleLines + Contract +
    PaperlessBilling + PaymentMethod + ten_fact, family = binomial(link = "logit"),
    data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8526  -0.7669  -0.3260   0.7845   3.0163

Coefficients:
                                       Estimate Std. Error z value Pr(>|z|)
(Intercept)                            -0.74181    0.11309  -6.560 5.39e-11 ***
SeniorCitizen1                          0.49711    0.08924   5.571 2.54e-08 ***
MultipleLinesYes                        0.54480    0.07942   6.859 6.92e-12 ***
ContractOne year                       -0.99512    0.11319  -8.791  < 2e-16 ***
ContractTwo year                       -2.14458    0.19290 -11.118  < 2e-16 ***
PaperlessBillingYes                     0.61432    0.07850   7.826 5.03e-15 ***
PaymentMethodCredit card (automatic) -0.06081    0.12225  -0.497  0.61887
PaymentMethodElectronic check           0.60351    0.10143   5.950 2.68e-09 ***
PaymentMethodMailed check              -0.33160    0.12085  -2.744  0.00607 **
ten_fact1-2years                       -0.82643    0.10225  -8.082 6.36e-16 ***
ten_fact2-3years                       -1.24092    0.12241 -10.138  < 2e-16 ***
ten_fact3-4years                       -1.06246    0.13372  -7.945 1.94e-15 ***
ten_fact4-5years                       -1.32037    0.14715  -8.973  < 2e-16 ***
ten_fact5-6years                       -1.56490    0.17421  -8.983  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 6525.6  on 5633  degrees of freedom
Residual deviance: 4957.6  on 5620  degrees of freedom
AIC: 4985.6

Number of Fisher Scoring iterations: 6
```

Therefore, we decided to perform stepwise selection with a full model to find the best performing model, and compare it with the previous models.

```
# 3- Perform stepwise selection with full model
OGdata_Step <- stepAIC(fit.full, direction = 'both')

#fit a model with best predictors defined in stepwise selection
Stepwise_model <- OGdata_Step

#Show fitted model
Stepwise_model
summary(Stepwise_model)
```

As we can see, the best model found with the stepwise regression has 15 predictors, and the lowest AIC 4729.7 than the previous two - full and reduced models.

```
> summary(Stepwise_model)

Call:
glm(formula = Churn ~ SeniorCitizen + Dependents + MultipleLines +
    InternetService + OnlineSecurity + DeviceProtection + TechSupport +
    StreamingTV + StreamingMovies + Contract + PaperlessBilling +
    PaymentMethod + MonthlyCharges + TotalCharges + ten_fact,
    family = binomial(link = "logit"), data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0411  -0.6697  -0.2920   0.6600   3.0670

Coefficients:
                                    Estimate Std. Error z value Pr(>|z|)
(Intercept)                         1.622e-01  3.011e-01   0.539 0.590032
SeniorCitizen1                      2.595e-01  9.352e-02   2.774 0.005532 **
DependentsYes                      -1.456e-01  9.185e-02  -1.585 0.112929
MultipleLinesYes                    3.961e-01  9.994e-02   3.964 7.38e-05 ***
InternetServiceFiber optic          1.445e+00  2.213e-01   6.529 6.64e-11 ***
InternetServiceNo                  -1.414e+00  1.984e-01  -7.125 1.04e-12 ***
OnlineSecurityYes                  -2.537e-01  1.018e-01  -2.492 0.012706 *
DeviceProtectionYes                 1.503e-01  9.289e-02   1.618 0.105595
TechSupportYes                     -1.726e-01  1.010e-01  -1.708 0.087671 .
StreamingTVYes                      4.766e-01  1.087e-01   4.383 1.17e-05 ***
StreamingMoviesYes                  5.004e-01  1.071e-01   4.672 2.98e-06 ***
ContractOne year                   -7.531e-01  1.199e-01  -6.279 3.40e-10 ***
ContractTwo year                   -1.638e+00  2.016e-01  -8.125 4.46e-16 ***
PaperlessBillingYes                 3.242e-01  8.344e-02   3.885 0.000102 ***
PaymentMethodCredit card (automatic) -4.401e-02  1.260e-01  -0.349 0.726933
PaymentMethodElectronic check       3.550e-01  1.058e-01   3.355 0.000793 ***
PaymentMethodMailed check          -5.166e-03  1.280e-01  -0.040 0.967815
MonthlyCharges                     -2.038e-02  6.633e-03  -3.072 0.002128 **
TotalCharges                       -1.081e-04  6.776e-05  -1.596 0.110541
ten_fact1-2years                   -8.210e-01  1.169e-01  -7.021 2.21e-12 ***
ten_fact2-3years                   -1.194e+00  1.665e-01  -7.172 7.42e-13 ***
ten_fact3-4years                   -9.331e-01  2.217e-01  -4.208 2.58e-05 ***
ten_fact4-5years                   -1.143e+00  2.871e-01  -3.981 6.85e-05 ***
ten_fact5-6years                   -1.278e+00  3.803e-01  -3.361 0.000776 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 6525.6  on 5633  degrees of freedom
Residual deviance: 4681.7  on 5610  degrees of freedom
AIC: 4729.7

Number of Fisher Scoring iterations: 6
```

To compare all three models, we run the ANOVA test. The results show us that the second model, which is our full model with all predictors, is significantly better. However, to choose the best model, we decided to do further analysis with all three models, compare their metrics, and made a decision based on these results.

```
> #Run the anova() function to compare three models
> anova(fit.reduced, fit.full, Stepwise_model, test="Chisq")
Analysis of Deviance Table

Model 1: Churn ~ SeniorCitizen + MultipleLines + Contract + PaperlessBilling +
    PaymentMethod + ten_fact
Model 2: Churn ~ gender + SeniorCitizen + Partner + Dependents + PhoneService +
    MultipleLines + InternetService + OnlineSecurity + OnlineBackup +
    DeviceProtection + TechSupport + StreamingTV + StreamingMovies +
    Contract + PaperlessBilling + PaymentMethod + MonthlyCharges +
    TotalCharges + ten_fact
Model 3: Churn ~ SeniorCitizen + Dependents + MultipleLines + InternetService +
    OnlineSecurity + DeviceProtection + TechSupport + StreamingTV +
    StreamingMovies + Contract + PaperlessBilling + PaymentMethod +
    MonthlyCharges + TotalCharges + ten_fact
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1      5620     4957.6
2      5606     4681.1 14  276.502   <2e-16 ***
3      5610     4681.7 -4   -0.521   0.9714
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Next, we showed the regression coefficients of exponentiated log-odds for all three models (odds), as they are easier for an explanation. As we can see, some predictors have higher coefficients in reduced models than in full and stepwise models, and vice versa. While, full and stepwise models, predictors have almost equal coefficients.

```
> #Show the regression coefficients of all models putting the results on an odds scale (odds)
> exp(coef(fit.reduced))
                        (Intercept)                       SeniorCitizen1                     MultipleLinesYes                    ContractOne year
                          0.4762526                           1.6439664                           1.7242577                           0.3696783
                    ContractTwo year                  PaperlessBillingYes PaymentMethodCredit card (automatic)         PaymentMethodElectronic check
                          0.1171177                           1.8483997                           0.9409975                           1.8285335
           PaymentMethodMailed check                     ten_fact1-2years                     ten_fact2-3years                     ten_fact3-4years
                          0.7177744                           0.4376100                           0.2891192                           0.3456043
                    ten_fact4-5years                     ten_fact5-6years
                          0.2670358                           0.2091085
> exp(coef(Stepwise_model))
                        (Intercept)                       SeniorCitizen1                       DependentsYes                     MultipleLinesYes
                          1.1761211                           1.2962263                           0.8645009                           1.4860767
          InternetServiceFiber optic                    InternetServiceNo                  OnlineSecurityYes                  DeviceProtectionYes
                          4.2416336                           0.2432303                           0.7759356                           1.1622147
                      TechSupportYes                      StreamingTVYes                  StreamingMoviesYes                    ContractOne year
                          0.8415010                           1.6106402                           1.6494470                           0.4709052
                    ContractTwo year                  PaperlessBillingYes PaymentMethodCredit card (automatic)         PaymentMethodElectronic check
                          0.1943909                           1.3828985                           0.9569406                           1.4261515
           PaymentMethodMailed check                     MonthlyCharges                        TotalCharges                     ten_fact1-2years
                          0.9948475                           0.9798304                           0.9998919                           0.4400126
                    ten_fact2-3years                     ten_fact3-4years                     ten_fact4-5years                     ten_fact5-6years
                          0.3030076                           0.3933129                           0.3188937                           0.2784781
> exp(coef(fit.full))
                        (Intercept)                          genderMale                       SeniorCitizen1                          PartnerYes
                          1.4634666                           0.9874012                           1.3011497                           0.9615454
                      DependentsYes                     PhoneServiceYes                     MultipleLinesYes          InternetServiceFiber optic
                          0.8802031                           1.1527927                           1.5523122                           5.2349043
                    InternetServiceNo                  OnlineSecurityYes                    OnlineBackupYes                DeviceProtectionYes
                          0.2020212                           0.8083150                           1.0826629                           1.2102800
                      TechSupportYes                      StreamingTVYes                  StreamingMoviesYes                    ContractOne year
                          0.8759217                           1.7464803                           1.7898153                           0.4705675
                    ContractTwo year                  PaperlessBillingYes PaymentMethodCredit card (automatic)         PaymentMethodElectronic check
                          0.1943447                           1.3796765                           0.9548363                           1.4253541
           PaymentMethodMailed check                     MonthlyCharges                        TotalCharges                     ten_fact1-2years
                          0.9934281                           0.9719686                           0.9998896                           0.4411208
                    ten_fact2-3years                     ten_fact3-4years                     ten_fact4-5years                     ten_fact5-6years
                          0.3039716                           0.3950730                           0.3206127                           0.2803007
```

Then, we created the confusion matrices for the train set prediction of all three models, the reduced model (on the left), the full model (in the middle), and the stepwise model (on the right). We can see that in the reduced model there are more TN and FN values than in the full and stepwise model. Although it seems that most of our predicted values match the actual values, we have a lot of FN values in the reduced model, which is more than the TP values. While in the last two models, FN values are less than TP values but still very close. This means that our reduced model mispredicted 851 customers attrition, as well as full and stepwise models mispredicted 726 customers attrition while, they did not churn.

We are 78.26% accurate in our reduced model classification, 80.17% in full, and 80.14% stepwise model classification. However, we also have a misclassification error rate of 21.74%, 19.83%, and 19.86% accordingly. Thus, the full and stepwise models have higher accuracy and lower misclassification error rate for almost 2% than the reduced model. For now, the full model seems to be better than other models

with 0.03% of higher accuracy than the stepwise model. But we should remember that the full model has 19 predictors while stepwise has 4 fewer predictors less.

Another important value to consider is Prevalence which is 0.2659 in all three models. This shows us that all models are at 26.6% prevalence of the positive class which is ‘Yes’ in the dataset. Thus, we have less churned customers rate and a higher rate of those who stayed.

```
> #Show the model accuracy with advanced
> #confusion matrix of the train set
> confusionMatrix(predicted.classes.min_reduced,
+                 train$Churn, positive = 'Yes')
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No  3762  851
       Yes  374  647

               Accuracy : 0.7826
                 95% CI : (0.7716, 0.7933)
    No Information Rate : 0.7341
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.3801

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.4319
            Specificity : 0.9096
         Pos Pred Value : 0.6337
         Neg Pred Value : 0.8155
             Prevalence : 0.2659
         Detection Rate : 0.1148
   Detection Prevalence : 0.1812
      Balanced Accuracy : 0.6707

       'Positive' Class : Yes
```

```
> #Show the model accuracy with advanced
> #confusion matrix of the train set
> confusionMatrix(predicted.classes.min_full,
+                 train$Churn, positive = 'Yes')
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No  3745  726
       Yes  391  772

               Accuracy : 0.8017
                 95% CI : (0.7911, 0.8121)
    No Information Rate : 0.7341
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4531

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.5154
            Specificity : 0.9055
         Pos Pred Value : 0.6638
         Neg Pred Value : 0.8376
             Prevalence : 0.2659
         Detection Rate : 0.1370
   Detection Prevalence : 0.2064
      Balanced Accuracy : 0.7104

       'Positive' Class : Yes
```

```
> #Show the model accuracy with advanced
> #confusion matrix of the train set
> confusionMatrix(predicted.classes.min_step,
+                 train$Churn, positive = 'Yes')
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No  3743  726
       Yes  393  772

               Accuracy : 0.8014
                 95% CI : (0.7907, 0.8117)
    No Information Rate : 0.7341
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4524

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.5154
            Specificity : 0.9050
         Pos Pred Value : 0.6627
         Neg Pred Value : 0.8375
             Prevalence : 0.2659
         Detection Rate : 0.1370
   Detection Prevalence : 0.2068
      Balanced Accuracy : 0.7102

       'Positive' Class : Yes
```

Then, we created the confusion matrices for the test set prediction of all three models. Here, all models have almost the same values in FN, FP, TN, and TP. But, in this case, we can see that as in the reduced model in the train set, the full and stepwise models now also have higher FN values than TP. As for the model's accuracy, the reduced model has the lowest accuracy as 78%, the full model has 80.2% accuracy, and the stepwise model has 80.34% of accuracy, which is higher than the full model for 0.14%. Thus, in the test set, the stepwise model has better performance than other models.

The Prevalence value is 0.2633 in all three models which is almost the same as it was in the train set. This shows us that all models are at 26.3% prevalence of the positive class which is ‘Yes’ in the dataset. Thus, we still have less churned customers rate and a higher rate of those who stayed.

```
> #Show the model accuracy with advanced
> #confusion matrix of the test set
> confusionMatrix(predicted.classes.min_test_reduced,
+                 test$Churn, positive = 'Yes')
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  940 212
       Yes  98 159

               Accuracy : 0.78
                 95% CI : (0.7574, 0.8014)
    No Information Rate : 0.7367
    P-Value [Acc > NIR] : 9.782e-05

                  Kappa : 0.3708

 Mcnemar's Test P-Value : 1.381e-10

            Sensitivity : 0.4286
            Specificity : 0.9056
         Pos Pred Value : 0.6187
         Neg Pred Value : 0.8160
             Prevalence : 0.2633
         Detection Rate : 0.1128
   Detection Prevalence : 0.1824
      Balanced Accuracy : 0.6671

       'Positive' Class : Yes
```

```
> #Show the model accuracy with advanced
> #confusion matrix of the test set
> confusionMatrix(predicted.classes.min_test_full,
+                 test$Churn, positive = 'Yes')
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  947 188
       Yes  91 183

               Accuracy : 0.802
                 95% CI : (0.7802, 0.8225)
    No Information Rate : 0.7367
    P-Value [Acc > NIR] : 6.053e-09

                  Kappa : 0.4428

 Mcnemar's Test P-Value : 9.064e-09

            Sensitivity : 0.4933
            Specificity : 0.9123
         Pos Pred Value : 0.6679
         Neg Pred Value : 0.8344
             Prevalence : 0.2633
         Detection Rate : 0.1299
   Detection Prevalence : 0.1945
      Balanced Accuracy : 0.7028

       'Positive' Class : Yes
```

```
> #Show the model accuracy with advanced
> #confusion matrix of the test set
> confusionMatrix(predicted.classes.min_test_step,
+                 test$Churn, positive = 'Yes')
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  950 189
       Yes  88 182

               Accuracy : 0.8034
                 95% CI : (0.7817, 0.8239)
    No Information Rate : 0.7367
    P-Value [Acc > NIR] : 2.817e-09

                  Kappa : 0.4447

 Mcnemar's Test P-Value : 1.873e-09

            Sensitivity : 0.4906
            Specificity : 0.9152
         Pos Pred Value : 0.6741
         Neg Pred Value : 0.8341
             Prevalence : 0.2633
         Detection Rate : 0.1292
   Detection Prevalence : 0.1916
      Balanced Accuracy : 0.7029

       'Positive' Class : Yes
```

To reflect on the other metrics that reflect the accuracy and reliability of the model we can look at the table below. It shows Sensitivity, Specificity, and Precision values for each of the models (Full model, Reduced model, Stepwise model).

	Sensitivity	**Specificity**	**Precision**
Full Model	0.4933	0.9123	0.6678
Reduced Model	0.4286	0.9056	0.6186
Stepwise Model	0.4906	0.9152	0.6740

As can be seen from the table, values for the Full Model and the Stepwise Model are very similar. The Sensitivity results show that more than 40% of all churned customers were predicted correctly. But the Full model has the highest value followed by the Stepwise Model, while the Reduced model has for 6.47% value less.

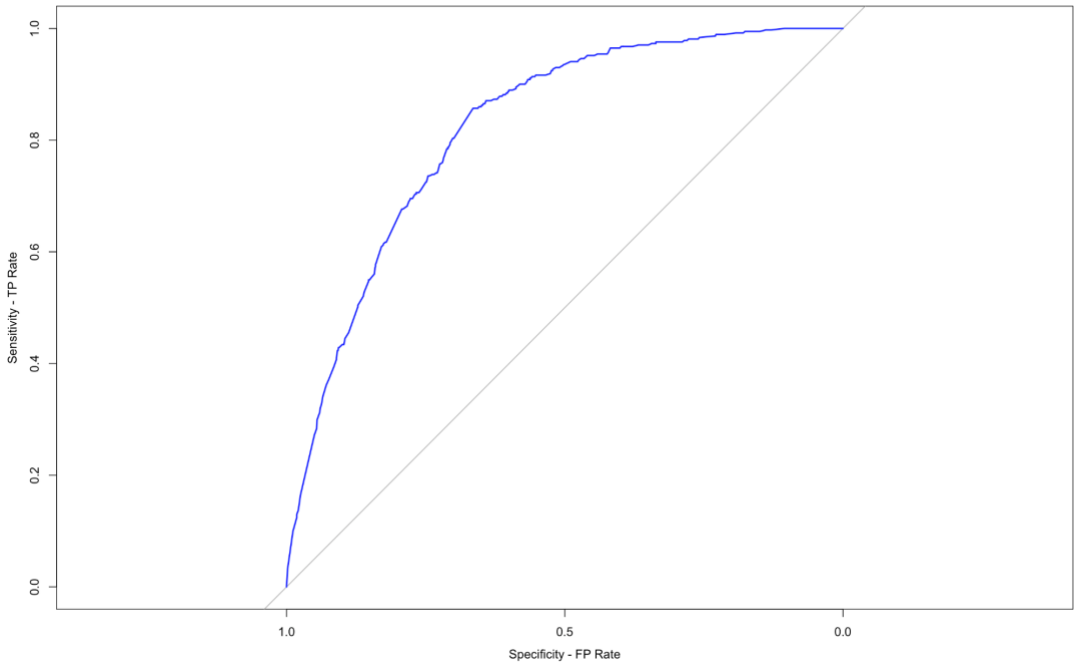
The Specificity results show that of all not churned customers, more than 90% were correctly predicted. But the Stepwise model has the highest value followed by the Full Model, while the Reduced model has for 0.96% value less.

T The Precision results show that more than 60% of customers that we labeled as churned are churned. But the Stepwise Model has the highest value followed by the Full Model, while the Reduced model has for 5.54% value less. From the analysis made earlier, it can be expected as those models are very similar in terms of accuracy as well.

Finally, we created data for the ROC curve to see the results of the R calculation and plotted the ROC curve that demonstrates Sensitivity which is our True Positive Rate versus Specificity which is False Positive Rate. The first model that will be analyzed is the Reduced Model.

```
> # Show ROC/AUC data
> myROC_reduced

Call:
roc.default(response = test$Churn, predictor = probabilities.test_reduced)

Data: probabilities.test_reduced in 1038 controls (test$Churn No) < 371 cases (test$Churn Yes).
Area under the curve: 0.8209
```

In other words, the graph demonstrates the summary of classifier performance. Ideally, if the classifiers show the curve closer to the top-left starting from one then the model considers performing a perfect prediction. We can see that our plot looks good but a bit rounded at the top. Anyways, with the set 50% threshold, the model looks to have good performance.

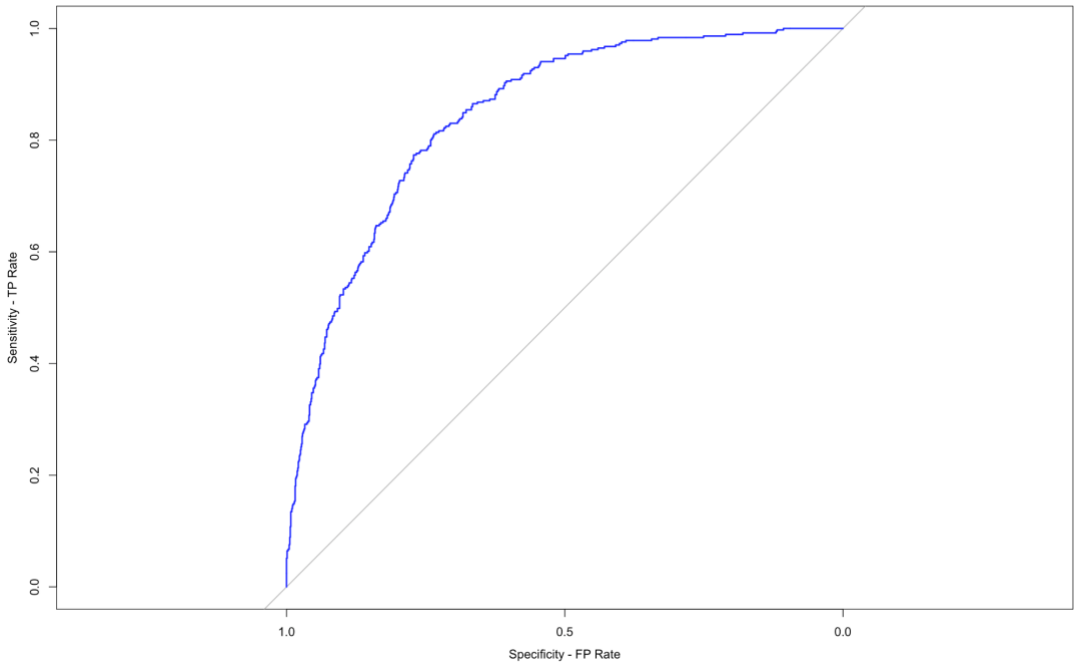


The Area Under the ROC curve helps us to identify the quality of model classification. Ideally, the AUC = 1 indicates a perfect classifier. We have calculated the area under the ROC curve (AUC) which is 0.8209. This means that the whole area under the ROC curve - a blue line in the plot - is 82% thus might be considered as a very good classifier. Thus, our model classifies almost 82% correctly.

The second model that will be analyzed is the Full Model.

```
> # Show ROC/AUC data
> myROC_full

Call:
roc.default(response = test$Churn, predictor = probabilities.test_full)

Data: probabilities.test_full in 1038 controls (test$Churn No) < 371 cases (test$Churn Yes).
Area under the curve: 0.845
```

The graph presented below demonstrates the summary of classifier performance. Similar to the one that was made for the Reduced Model, we can conclude that the model has good performance.

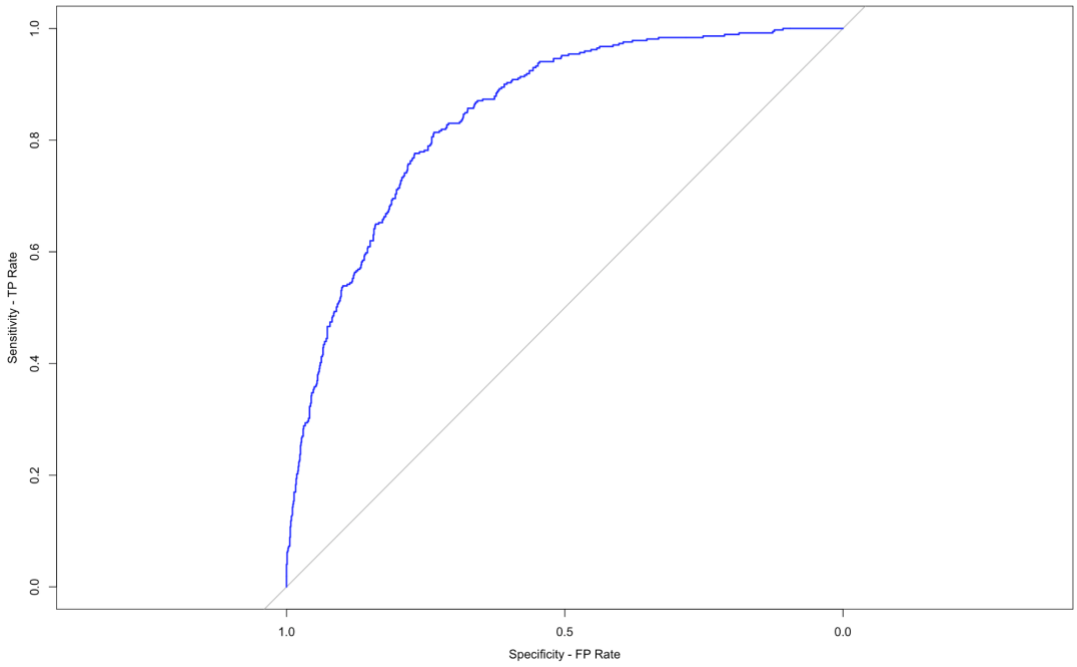


The Area Under the ROC curve helps us to identify the quality of model classification. Ideally, the AUC = 1 indicates a perfect classifier. We have calculated the area under the ROC curve (AUC) which is 0.845, which is higher than in the Reduced Model. This means that the whole area under the ROC curve - a blue line in the plot - is 84.5% thus might be considered as a very good classifier. Thus, our model classifies almost 85% correctly.

The third model that will be analyzed is the Stepwise Model.

```
> # Show ROC/AUC data
> myROC_step

Call:
roc.default(response = test$Churn, predictor = probabilities.test_step)

Data: probabilities.test_step in 1038 controls (test$Churn No) < 371 cases (test$Churn Yes).
Area under the curve: 0.8454
```

The graph presented below demonstrates the summary of classifier performance. Similar to the one that was made for the Stepwise Model, we can conclude that the model has good performance.



The Area Under the ROC curve helps us to identify the quality of model classification. Ideally, the AUC = 1 indicates a perfect classifier. We have calculated the area under the ROC curve (AUC) which is 0.8454, which is higher than in the Reduced Model. This means that the whole area under the ROC curve - a blue line in the plot - is 84.5% thus might be considered as a very good classifier. Thus, our model classifies almost 85% correctly.

From the analysis of the models, it was found that the Reduced Model with 6 significant predictors, despite good indicators, and high performance, is the most underperformed model out of all the three models tested. Although the Full and Stepwise models have almost the same metric values, the Accuracy, Specificity, Precision, and AUC values are slightly prevailing in the Stepwise model. On top of

that, the model uses 16 variables to predict <Churn>, whereas a Full Model, which performs almost similarly, uses 19 variables. Thus, the Stepwise Model with 15 predictors, namely:

SeniorCitizen + Dependents + MultipleLines + InternetService + OnlineSecurity + DeviceProtection + TechSupport + StreamingTV + StreamingMovies + Contract + PaperlessBilling + PaymentMethod + MonthlyCharges + TotalCharges + ten_fact seems to perform better than other two models.

LASSO Regression Model

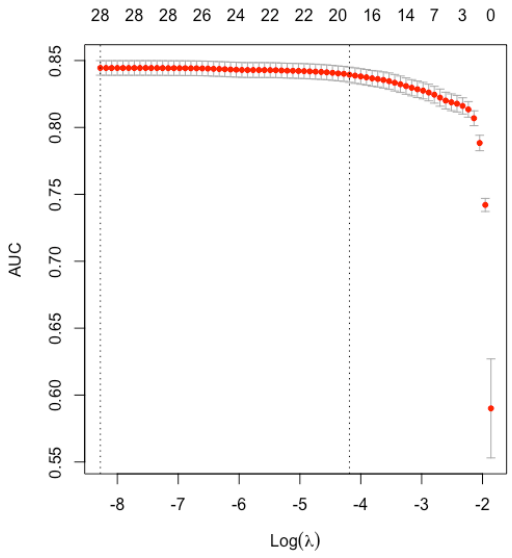
The second model we applied is the LASSO regression model which can reduce insignificant coefficients of predictors to zero to make the model flexible.

As our response variable, <Churn>, is a binary variable, in the LASSO regularization process, we used the 'family' of "binomial", and the measurement of lambda is 'AUC' (area under the ROC curve). We used the glmnet() function that uses K-fold cross-validation with 10 folds.

```
# Use cv.glmnet function to cross validate to find the best lambda values
# The cross validation method I chose is K-fold and here, K is 10.
CV.L1 <- cv.glmnet(trainX, trainY, alpha = 1, nfolds = 10, family="binomial", type.measure='auc')
```

The following screenshots show lambda values (minimum and one-standard-error lambda value) and the plot of AUC versus the log of lambda:

```
> # lambda.min
> CV.L1$lambda.min
[1] 0.00143047
> # log of lambda.min
> log(CV.L1$lambda.min)
[1] -6.549752
> # one-standard-error lambda
> CV.L1$lambda.1se
[1] 0.01464131
> # log of one-standard-error lambda
> log(CV.L1$lambda.1se)
[1] -4.223908
```



The log of minimum lambda is -6.549752 and the log of one-standard-error lambda is -4.223908. They are represented by 2 dotted lines on the graph. The line on the left is a log of minimum lambda that has 28 non-zero-coefficient predictors and the dotted line on the right depicts the log of one-standar-error lambda having 21 non-zero coefficient predictors.

We can also notice that the AUC value for the log of minimum lambda is 0.84 whereas, for log of one-standard-error lambda, it is 0.83. In other words, the performance of the LASSO regression model will reduce marginally while the flexibility of the model will improve significantly.

In the screenshots below we can see the coefficients using LASSO regularization regression with minimum lambda (the left one) and with one-standard-error lambda (the right one). The number of predictors will reduce with the increase of lambda value:

minimum lambda has 17 predictors while one-standard-error lambda has only 12 predictor variables.

```
> # Lambda.min
> modL1.min <- glmnet(trainX, trainY, alpha = 1, lambda = CV.L1$lambda.min, family="binomial")
> # View the coefficients
> coef(modL1.min)
35 x 1 sparse Matrix of class "dgCMatrix"
                                          s0
(Intercept)                       -0.4480653249
genderMale                         .
SeniorCitizen1                     0.2476123504
PartnerYes                        -0.0358185209
DependentsYes                     -0.1245859214
PhoneServiceYes                   -0.3320606900
MultipleLinesNo phone service      .
MultipleLinesYes                   0.2597403968
InternetServiceFiber optic         0.9645263397
InternetServiceNo                 -0.8876184593
OnlineSecurityNo internet service  .
OnlineSecurityYes                 -0.3330258638
OnlineBackupNo internet service    .
OnlineBackupYes                   -0.0356393696
DeviceProtectionNo internet service .
DeviceProtectionYes                0.0227073722
TechSupportNo internet service     .
TechSupportYes                    -0.2445421275
StreamingTVNo internet service     .
StreamingTVYes                     0.2717163357
StreamingMoviesNo internet service .
StreamingMoviesYes                 0.2946733687
ContractOne year                  -0.7465538859
ContractTwo year                  -1.6021575006
PaperlessBillingYes                0.3082400099
PaymentMethodCredit card (automatic) -0.0279422990
PaymentMethodElectronic check      0.3664806875
PaymentMethodMailed check          .
MonthlyCharges                     .
TotalCharges                      -0.0001797275
ten_fact1-2years                  -0.6824640784
ten_fact2-3years                  -0.9713353077
ten_fact3-4years                  -0.6359742730
ten_fact4-5years                  -0.7582023653
ten_fact5-6years                  -0.7902625940
```

```
> # Lambda.1se
> modL1.1se <- glmnet(trainX, trainY, alpha = 1, lambda = CV.L1$lambda.1se, family="binomial")
> # View the coefficients
> coef(modL1.1se)
35 x 1 sparse Matrix of class "dgCMatrix"
                                          s0
(Intercept)                       -0.8728020755
genderMale                         .
SeniorCitizen1                     0.1337723435
PartnerYes                         .
DependentsYes                     -0.0763959078
PhoneServiceYes                    .
MultipleLinesNo phone service      .
MultipleLinesYes                   .
InternetServiceFiber optic         0.9259920781
InternetServiceNo                 -0.7546033361
OnlineSecurityNo internet service  .
OnlineSecurityYes                 -0.2259158141
OnlineBackupNo internet service    .
OnlineBackupYes                    .
DeviceProtectionNo internet service .
DeviceProtectionYes                .
TechSupportNo internet service     .
TechSupportYes                    -0.0981116967
StreamingTVNo internet service     .
StreamingTVYes                     0.0918355091
StreamingMoviesNo internet service .
StreamingMoviesYes                 0.1125008617
ContractOne year                  -0.6276121817
ContractTwo year                  -1.2310110972
PaperlessBillingYes                0.2134314083
PaymentMethodCredit card (automatic) .
PaymentMethodElectronic check      0.4235079318
PaymentMethodMailed check          .
MonthlyCharges                     .
TotalCharges                      -0.0002186572
ten_fact1-2years                  -0.0972909321
ten_fact2-3years                  -0.2450645630
ten_fact3-4years                   .
ten_fact4-5years                   .
ten_fact5-6years                   .
```

In the screenshots below we show the confusion matrices of model fitting on the training set with minimum lambda (the left one) and with one-standard-error lambda (the right one).

Confusion matrix of minimum lambda shows 3762 True Negatives, 767 True Positives, 374 False Positive and 731 False Negative and the accuracy of the model is 0.803 whereas, for one-standard-error lambda , the confusion matrix shows 3817 True Negatives, 671 True Positives, 319 False Positive and 827 False Negative and the accuracy of the model is 0.796.

```
> confusionMatrix(predicted.train.min, trainY, positive = 'Yes')
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No  3762  731
       Yes  374  767

               Accuracy : 0.8039
                 95% CI : (0.7933, 0.8142)
    No Information Rate : 0.7341
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4563

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.5120
            Specificity : 0.9096
         Pos Pred Value : 0.6722
         Neg Pred Value : 0.8373
             Prevalence : 0.2659
         Detection Rate : 0.1361
   Detection Prevalence : 0.2025
      Balanced Accuracy : 0.7108

       'Positive' Class : Yes
```

```
> confusionMatrix(predicted.train.1se, trainY, positive = 'Yes')
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No  3817  827
       Yes  319  671

               Accuracy : 0.7966
                 95% CI : (0.7858, 0.807)
    No Information Rate : 0.7341
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4158

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.4479
            Specificity : 0.9229
         Pos Pred Value : 0.6778
         Neg Pred Value : 0.8219
             Prevalence : 0.2659
         Detection Rate : 0.1191
   Detection Prevalence : 0.1757
      Balanced Accuracy : 0.6854

       'Positive' Class : Yes
```

We see the following screenshots of the confusion matrices of model fitting on the test set with minimum lambda (the left one) and with one-standard-error lambda (the right one).

For the test set, confusion matrix minimum lambda shows 951 True Negatives, 179 True Positives, 87 False Positive and 192 False Negative and the accuracy of the model is 0.802 whereas, for one-standard-error lambda , the confusion matrix shows 963 True Negative, 163 True Positives, 75 False Positive and 208 False Negative and the accuracy of the model is 0.796.

```
> confusionMatrix(predicted.test.min, testY, positive = 'Yes')
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  951 192
       Yes  87 179

               Accuracy : 0.802
                 95% CI : (0.7802, 0.8225)
    No Information Rate : 0.7367
    P-Value [Acc > NIR] : 6.053e-09

                  Kappa : 0.4385

 Mcnemar's Test P-Value : 4.775e-10

            Sensitivity : 0.4825
            Specificity : 0.9162
         Pos Pred Value : 0.6729
         Neg Pred Value : 0.8320
             Prevalence : 0.2633
         Detection Rate : 0.1270
   Detection Prevalence : 0.1888
      Balanced Accuracy : 0.6993

       'Positive' Class : Yes
```

```
> confusionMatrix(predicted.test.1se, testY, positive = 'Yes')
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  963 208
       Yes  75 163

               Accuracy : 0.7991
                 95% CI : (0.7773, 0.8198)
    No Information Rate : 0.7367
    P-Value [Acc > NIR] : 2.652e-08

                  Kappa : 0.4149

 Mcnemar's Test P-Value : 4.275e-15

            Sensitivity : 0.4394
            Specificity : 0.9277
         Pos Pred Value : 0.6849
         Neg Pred Value : 0.8224
             Prevalence : 0.2633
         Detection Rate : 0.1157
   Detection Prevalence : 0.1689
      Balanced Accuracy : 0.6835

       'Positive' Class : Yes
```

The accuracy, sensitivity, specificity, and precision for the logistic regression with minimum lambda and one-standard-error lambda fitting on the training set and test set are shown in the following chart:

	accuracy	sensitivity	specificity	precision
min lambda on train	0.8039	0.5120	0.9096	0.6722
min lambda on test	0.802	0.4825	0.9162	0.6729
1SE lambda on train	0.7966	0.4479	0.9229	0.6778
1SE lambda on test	0.7991	0.4394	0.9277	0.6849

Minimum lambda has the highest accuracy on test with 80.2% accuracy of the model performance whereas the accuracy of model with one-standard-error lambda is marginally lower at 0.7991, which means the model predicted the correct outcome 79.91% times.

The test set of models with one-standard-error lambda has the highest precision which means 68.49% proportion of the True positive values were predicted correctly.

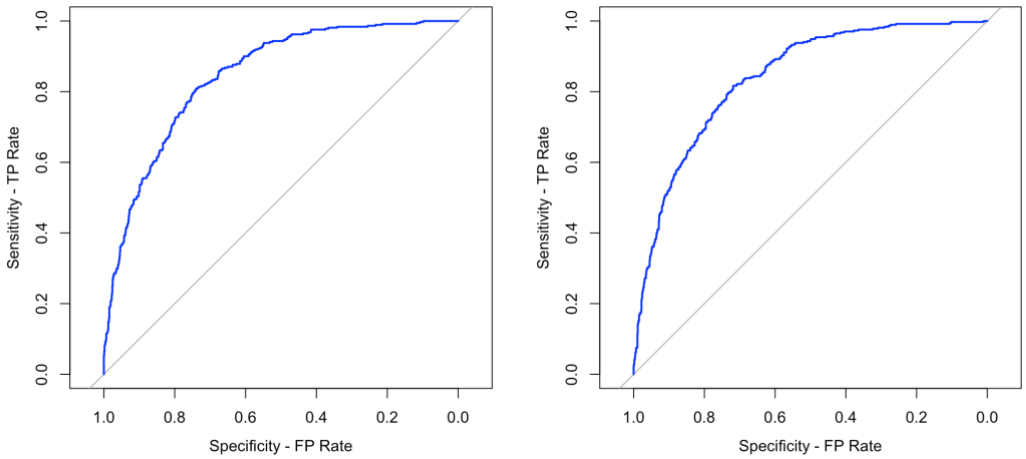
The specificity of the model with one-standard-error lambda is also the highest which means that model predicted the negative values with 92.77% accuracy.

Within the test set, the model with one-standard-error lambda has the highest sensitivity with 0.4825 meaning 48.25% times the model predicted the observations from positive cases correctly.

The following screenshot shows the AUC values in different model fitting. We can observe that the model with minimum lambda has better performance on the area under the ROC curve. AUROC plots the False Positive Rate (Specificity) and True Positive Rate (Sensitivity)

	model	AUC
1	Minimum Lambda on Train	0.8464769
2	Minimum Lambda on Test	0.8453199
3	1se Lambda on Train	0.8399667
4	1se lambda on Test	0.8396915

The following screenshots respectively show the ROC curve for the model with minimum lambda fitting on the test set and the model with one-standard-error lambda fitting on the test set:



On comparing the AUC of models with minimum lambda and one-standard-error lambda, we find that the model with minimum lambda has a higher accuracy rate of 84.53% in comparison with one-standard-error lambda with 83.96% accuracy.

Conclusion

In this project, we worked on building a predictive model to predict the Churn rate for Telco, so we analyzed all relevant predictive customer information to determine effects on Churn and tried to develop targeted customer retention. We chose these Logistic Regression models and LASSO Regression models to understand key attributes that impact customers to leave within the last month and predict their behavior. We built a full model with all predictive variables and gained lower AIC compared to a reduced model with 6 significant variables. Further, we performed Stepwise selection with a full model to find a better-performed model, compared with previous models. We acquired a better model with stepwise regression with 15 predictors with the lowest AIC compared to full and reduced logistic models. Furthermore, to reach a better-performing model, we made some tests.

From the result of the ANOVA test, we acquired better performance from a full model with lower variance. Despite that, we created a confusion matrix on training data set for all three models and as the outcome of these matrices, we observed that most of our model's predictions fit the actual values when reduced model made in more incorrect prediction with 851 False Negative values, which is higher than True Positive values (647). However, the full and stepwise model provided us less false negative (FN) values compared to true positive (TP) values. In a word, the last two models mispredicted less in converse to a reduced model with

significant variables. When looking at the accuracy of models, we got less accuracy or higher error rate in reduced models while full and stepwise models provided higher accuracy or lower error rate for about 2% comparison on the reduced model. We should pay attention to having the same Prevalence within the three models, which explains we have less churn rate compared with a higher rate of customers who are staying in Telco. We also looked through the performance of models on the testing set and gained similar results. It is seen that full and stepwise models are in similar predictions, only the Stepwise model's accuracy, specificity, precision, and AUC are fewer higher than full of 15 and 19 predictors, respectively.

As part of the ROC curve and AUC values result, the Reduced model classifies almost 82% correctly while Stepwise, as well as Full model, classifies almost 85%. It is interesting that the Reduced Model with 6 significant predictors, despite good indicators, high performance, and highest Specificity value, is an underperformed model rather than the other two models.

For the LASSO regression model, as LASSO regularization can reduce the predictors' coefficients to zero, it makes the model more flexible. Lambda.min with the highest accuracy (80.2%) whereas the accuracy of lambda.1se model is gradually lower at 79.91, which indicates the model predicted the correct outcome 79.91% on the testing data set. The 68.49% precision of lamda.1se on the test set shows the highest prediction which means almost 69% true positive values were predicted correctly while 92.77% accuracy of negative values was predicted. However, on the test set, lambda.min shows the highest sensitivity with 48.25% that indicates model 48.25% predicted observations from positive cases in the correct. Moreover, from the result of AUC, we conclude that lambda.min on the train and test data set has a better performance compared with lamda.1se.

Recommendations

Evaluation of the models within each method is very important to avoid possible mistakes and overfit problems. This will help us to have a better prediction model. In this case, we assume that we have reached the goal of this assignment and built two effective models using two methods: logistic regression, and LASSO regression with good performances. As we can see, both Stepwise and Lambda.min

models built with different methods have 14 mutual predictors that impact on response variable <Churn> that can help us to predict the customers' behavior effectively and accurately if they will leave the Telco company or not. They are: "SeniorCitizen", "Dependents", "MultipleLines", "InternetService", "OnlineSecurity", "DeviceProtection", "TechSupport", "StreamingTV", "StreamingMovies" , "Contract", "PaperlessBilling", "PaymentMethod", "TotalCharges", and "ten_fact".

They are significant variables that have an impact on or are more related to the Churn behavior of Telco customers. The difference between the two models is that the Lasso model has three more significant predictors that affect the customers' churn rate, as "Partner", "PhoneService", and "OnlineBackup", as well as does not have the predictor "MonthlyCharges".

References:

BlastChar. (2018, February 23). *Telco customer churn*. Kaggle. Retrieved January 26, 2022, from https://www.kaggle.com/blastchar/telco-customer-churn

Bluman, A. G. (2014). *Elementary Statistics*.

Linear, Lasso, and Ridge Regression with R | Pluralsight. (n.d.). Pluralsight | The Tech Workforce Development Company. Retrieved February 10, 2022, from https://www.pluralsight.com/guides/linear-lasso-and-ridge-regression-with-r