

**Logistic Regression**

José Gerardo Huerta De Rubín

José Francisco Rico Moreno

**Department:** Industrial Engineering

**Course:** IN99290 Statistical Inference and Data Science II **Professor:** Marzieh Khakifirooz, PhD.

**Date:** October 04, 2023

**Abstract:** Logistic regression is a type of statistical model that is used to predict binary outcome’s probability. In this work, a logistic regression model was developed and tested to analyze a dataset concerning the churn rate of customers of a certain company (and containing different variables that may be relevant for the churn rate). This allowed to understand the situation and give some insights to the organization.

**1. Introduction**

Logistic regression is a statistical model used to predict a binary outcome's probability. The outcome can be any categorical variable with two possible values, such as "success" or "failure", "yes" or "no".

This regression method is a type of generalized linear model (GLM). GLMs are a broad class of statistical models that can be used to model various types of data, including continuous, binary, and categorical data.

Logistic regression models the relationship between the independent variables and the probability of the binary outcome. The independent variables can be any variable, including continuous, categorical, and interaction terms.

Once the logistic regression model has been trained, it can predict the probability of the binary outcome for new observations. This can be done by simply plugging the values of the independent variables into the model and calculating the predicted probability.

Here are some examples of how logistic regression can be used:

* Predicting whether a customer will churn.
* Predicting whether a patient has a particular disease.
* Predicting whether a loan applicant will default on their loan.
* Predicting whether a student will pass an exam.

Regarding this issue, we possess categorical and continuous data on the behavior of various clients of a potential internet services company. Our primary goal is to predict whether clients will churn.

**2. Procedures and Results**

## Data

The information that is provided about the customers is an ID, their gender, whether or not they are Senior citizens, if they have not partnered and dependents, the tenure of the service, if they have phone services, as well as the kind of service they have, their internet services, payment methods, and the charges (both monthly and total). The data is resumed in the next tables as well as its category.

|  |  |
| --- | --- |
| Data | Type of data |
| ID | Categorical |
| Gender | Categorical/binary |
| Senior Citizen | Categorical/binary |
| Partner | Categorical/binary |
| Dependents | Categorical/binary |
| Tenure | Continuous |
| Phone Services | Categorical/binary |
| Multiple Lines | Categorical |
| Internet Service | Categorical |
| Online Security | Categorical |
| Online Backup | Categorical |
| Device Protection | Categorical |
| Tech Support | Categorical |
| Streaming TV | Categorical |
| Streaming movies | Categorical |
| Contract | Categorical |
| Paperless Billing | Categorical/binary |
| Payment Method | Categorical |
| Monthly Charges | Continuous |
| Total Charges | Continuous |
| Churn | Categorical/binary |

Table Data type

1. Data Preparation

The first step was to treat the data to be analyzed. For this, the following code was used to import and visualize the data. This allowed to import the data from a local address and view it as in figure 1.

# Load the Customer Churn Dataset

customer\_churn\_dataset <- read.csv('C:\\Users\\JGHDR\\Documents\\MAESTRÍA EN CIENCIAS DE LA INGENIERÍA\\3er semestre AgoDic2023\\Tópicos selectos\\WA\_Fn-UseC\_-Telco-Customer-Churn.csv')

View(customer\_churn\_dataset)

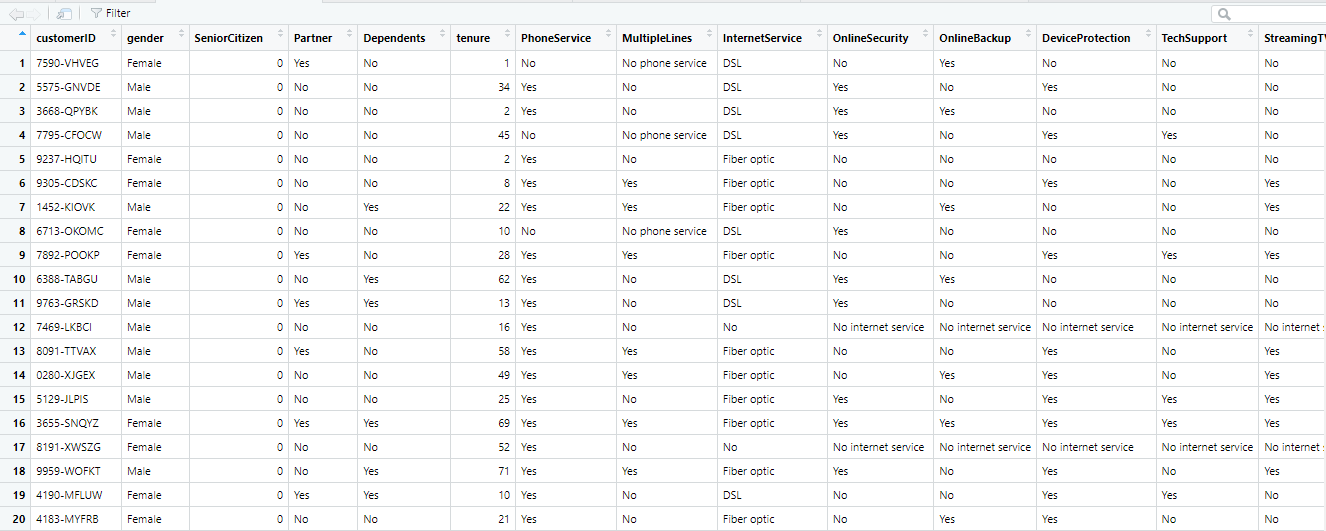


Figure . Dataset shown in RStudio before treatment.

After that, the data was treated. First, the missing values were removed, and the binary variables were defined as such. For this, all variables containing “Yes”/”No” answers were identified and explicitly mentioned, and using an ifelse() function, were turned into numerical binary variables. The rest of the categorical variables were transformed into numerical creating dictionaries (and defining a number for each category); these variables were converted into numerical and then, all variables were treated and defined as factors to be analyzed in latter steps. The new dataset was overwritten (the result can be seen in Figure 2.) and, to verify the type of variables, the str() function was used (as seen in Figure 3). Sections of the code can be seen in the following lines; the full code can be found in the repository of this document.

# Identify the binary variables

binary\_variables <- c('Partner', 'Dependents', 'PhoneService', 'MultipleLines', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies','PaperlessBilling', 'Churn')

# Convert the binary variables to numerical variables using the ifelse() function

for (binary\_variable in binary\_variables) {

customer\_churn\_dataset[, binary\_variable] <- ifelse(customer\_churn\_dataset[, binary\_variable] == 'Yes', 1, 0)

}

# Create a dictionary to map the levels of the categorical variables to integer values

payment\_method\_mapping <- list(

"Electronic check" = 1,

"Mailed check" = 2,

"Bank transfer (automatic)" = 3,

"Credit card (automatic)" = 4

)

# Convert the variables to numerical using factors

customer\_churn\_dataset$gender <- as.numeric(factor(customer\_churn\_dataset$gender, levels = names(gender\_mapping)))

customer\_churn\_dataset$InternetService <- as.numeric(factor(customer\_churn\_dataset$InternetService, levels = names(internet\_service\_mapping)))

#MAKE ALL VARIABLES FACTORS

customer\_churn\_dataset$gender <- as.factor(customer\_churn\_dataset$gender)

customer\_churn\_dataset$SeniorCitizen <- as.factor(customer\_churn\_dataset$SeniorCitizen)

# Save the converted dataset

write.csv(customer\_churn\_dataset, 'Customer\_Churn\_Dataset\_Numerical.csv', row.names = FALSE)

View(customer\_churn\_dataset)

str(customer\_churn\_dataset)

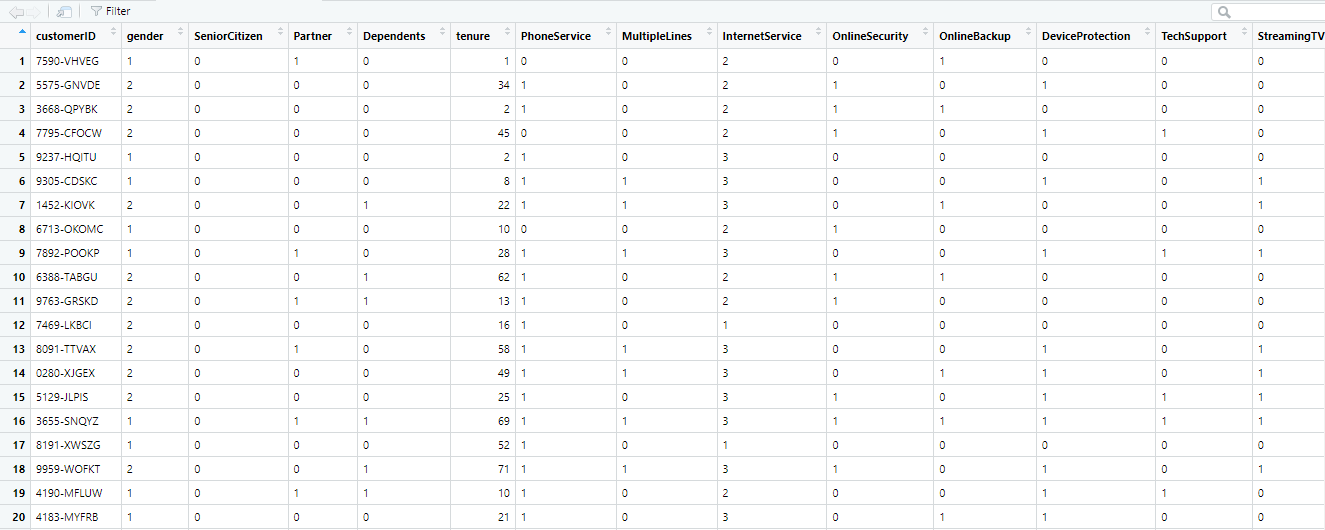


Figure . Dataset shown in RStudio after treatment (only factors and numbers).

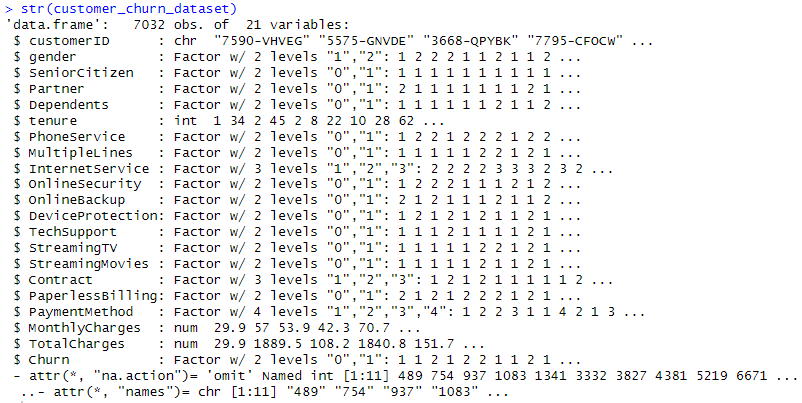


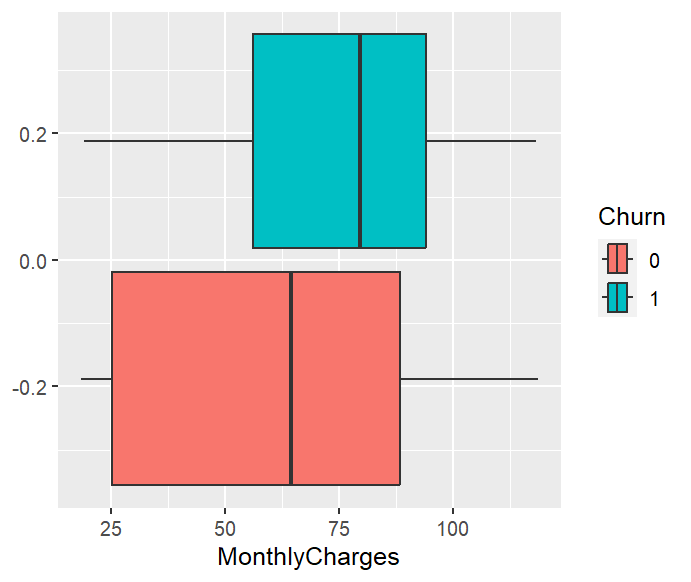
Figure . Result of the structure function in the console of RStudio, where can be seen that most variables are now factors.

1. Exploratory Data Analysis

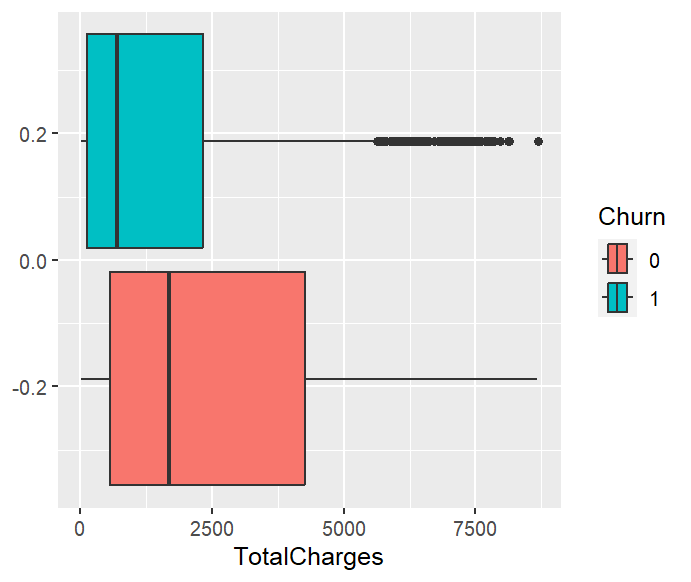
A) VISUAL INFORMATION

Since the majority of the data is categorical, much of the visual information can be resumed as most of the variables have an imbalanced result when categorizing it into churn or not churn. Appendix A shows different distribution plots of the categorical data in constant with the response variable Churn. From them, we can conclude that some variables, such as Internet service and tenure, are variables that might influence in the response variables due to the variability between the categorical data and the response variable. Summary statistics of both continuous and categorical data are displayed in Figure 5.

By analyzing graphs 1 and 2, it appears that the monthly service charges could be a significant factor in determining whether clients will churn. This is because the clients who churned have a lower variance and higher density in the higher cost bracket. However, total charges do not seem to be a critical variable as both categories have lower costs. Additionally, the "Yes churn" category has a high number of outliers.



Graph. Monthly charges distribution by response variable



Graph. Total charges grouped by response variable.

Once the data was analyzed, some initial visual explorations were done. In Figure 4, a histogram can be seen for monthly charges (in the train set). The charges are concentrated in two points, with most of them being lower but, after a certain initial point, a gaussian type of distribution can be seen.

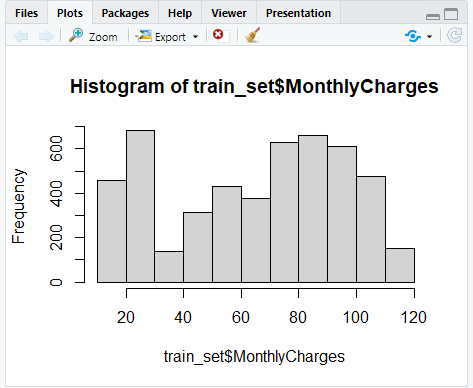
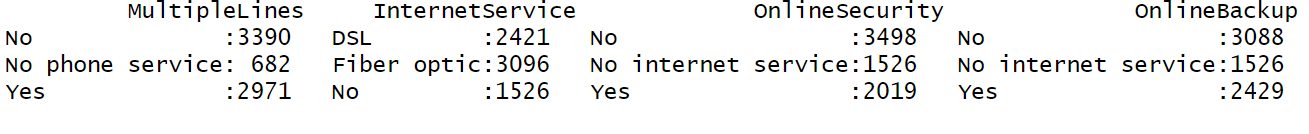


Figure . Histogram of Monthly Charges.

B) SUMMARY STATISTICS

Texto

Descripción generada automáticamente



Texto

Descripción generada automáticamente con confianza baja

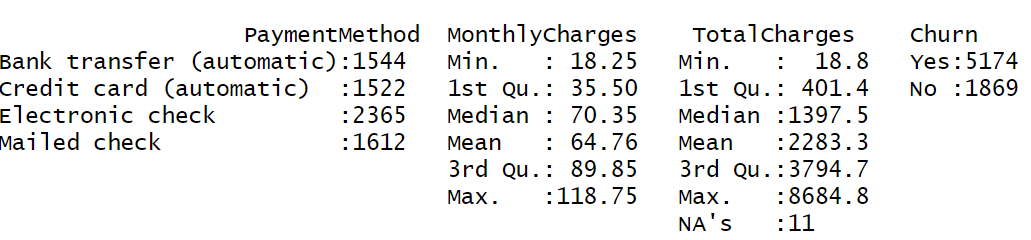


Figure Summary statistics for each variable of the dataset. Retrieved from R

C) CATEGORICAL CODES

In order to manipulate the data, categorical variables were turned numerical considering the following codes:

a) Gender variable

|  |  |
| --- | --- |
| Name of the category | Number assigned |
| Female | 0 |
| Male | 1 |

b) Internet service variable

|  |  |
| --- | --- |
| Name of the category | Number assigned |
| No | 0 |
| DSL | 1 |
| Fiber optic | 2 |

c) Contract variable

|  |  |
| --- | --- |
| Name of the category | Number assigned |
| Month-to-month | 1 |
| One year | 2 |
| Two year | 3 |

d) Payment method variable

|  |  |
| --- | --- |
| Name of the category | Number assigned |
| Electronic check | 1 |
| Mailed check | 2 |
| Bank transfer (automatic) | 3 |
| Credit card (automatic) | 4 |

Considering the previous codes, the variables can be summarized as:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Variable | 0 | 1 | 2 | 3 | 4 |
| Gender | Female | Male |  |  |  |
| Senior Citizen | No | Yes |  |  |  |
| Partner | No | Yes |  |  |  |
| Dependents | No | Yes |  |  |  |
| Tenure | Numerical | | | | |
| Phone Service | No | Yes |  |  |  |
| Multiple Lines | No | Yes |  |  |  |
| Internet Service | No | DSL | Fiber optic |  |  |
| Online Security | No | Yes |  |  |  |
| Online Backup | No | Yes |  |  |  |
| Device Protection | No | Yes |  |  |  |
| Tech Support | No | Yes |  |  |  |
| Streaming TV | No | Yes |  |  |  |
| Streaming Movies | No | Yes |  |  |  |
| Contract |  | Month-to-month | One year | Two year |  |
| Paperless Billing | No | Yes |  |  |  |
| Payment Method |  | Electronic check | Mailed check | Bank transfer (automatic) | Credit card (automatic) |
| Monthly Charges | Numerical | | | | |
| Total Charges | Numerical | | | | |
| Churn | No | Yes |  |  |  |

1. Feature selection

First, the data was split into training and testing sets (considering 70% for training and the rest for testing). Once the data was split, and ensured it was numeric, a linear model was first declared, and the results showed it was not adequate (figure 6). Once this was proven, a logistic regression model was developed.

# Build a logistic regression model to predict customer churn

m2 <- glm(Churn ~ gender+SeniorCitizen+Partner+Dependents+tenure+PhoneService+MultipleLines+InternetService+OnlineSecurity+OnlineBackup+DeviceProtection+TechSupport+StreamingTV+StreamingMovies+Contract+PaperlessBilling+PaymentMethod+MonthlyCharges+TotalCharges, data = train\_set, family=binomial)

summary(m2)

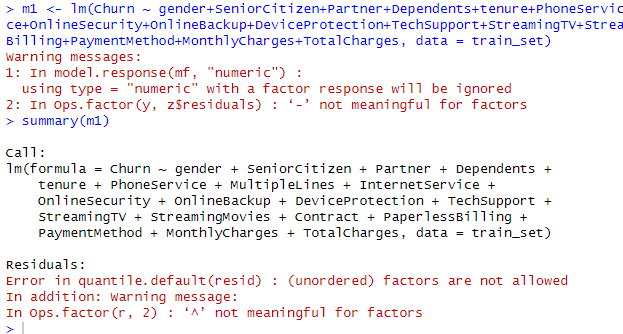


Figure . Results (in console) of the linear model (errors demonstrate it is not adequate).

A general linear model was created; its summary is shown in figure 7. As some variables were found to be non-significant, the model was improved through iterations until the best one was obtained. In each iteration, the variable with the highest p-value was removed until the AIC indicated that the model was no longer better than the previous one. The best model was found on the eighth iteration; its summary is shown in figure 8. This model helped identify the only relevant variables: Dependents, tenure, Multiple Lines, Internet Service, Online Security, Tech Support, Streaming TV, Streaming Movies, contracts, Paperless Billing, Payment Method, Monthly Charges, and Total Charges.

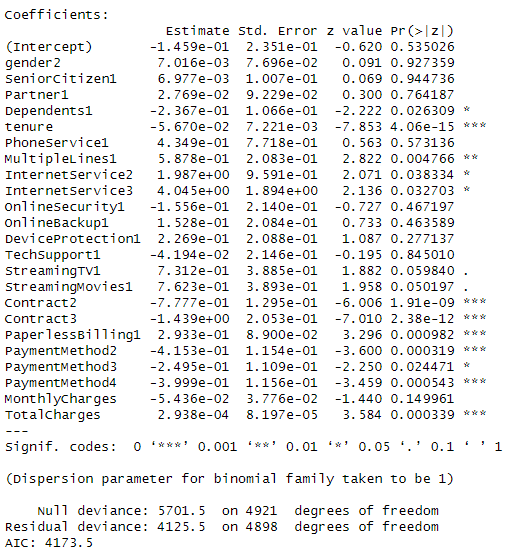


Figure . Summary (in console) of the first general linear model.

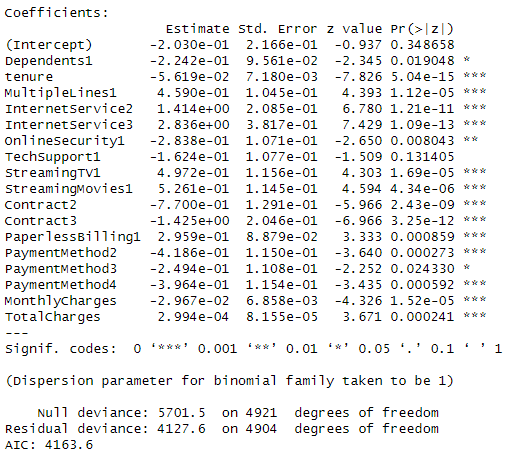


Figure . Summary (in console) of the improved general linear model.

1. Logistic Regression Modeling

Once the model was defined, it was trained. For this, the library “caret” was used, and a new model was defined, and used to make predictions. The results of the logistic regression model (trained) can be seen in figure 9.

library(caret)

library(pROC)

newmodel <- train(

form = Churn ~ SeniorCitizen+Dependents+tenure+MultipleLines+InternetService+OnlineSecurity+TechSupport+StreamingTV+StreamingMovies+Contract+PaperlessBilling+PaymentMethod+MonthlyCharges+TotalCharges ,

data = train\_set,

trControl = trainControl(method = "cv", number = 5),

method = "glm",

family = "binomial"

)

# Make predictions on the test set

predictions <- predict(newmodel, newdata = test\_set, type = "raw")

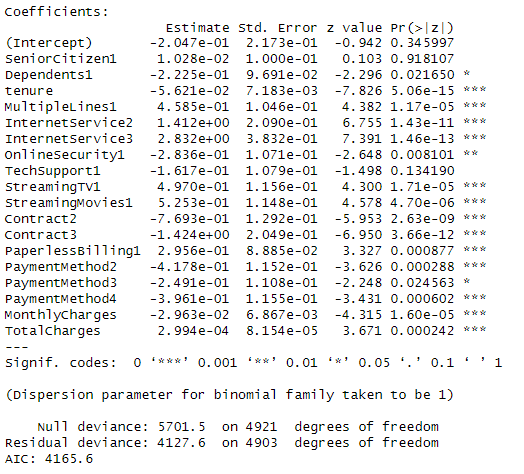


Figure . Summary (in console) of the trained logistic regression model.

1. Model Evaluation

Once the model was developed and tested, its performance was evaluated. For this, the accuracy, precision, recall and F1-score were obtained using their corresponding functions; the result can be seen in figure 10. As shown in the results, the model is a good one, as all these metrics are over 0.70 and close to 1, which means that the model can correctly predict most of the outcomes, as well as resulting in most predictions being correct, it is also able to identify most of the positive cases. It has a good balance between precision and recall. Similarly, the ROC AUC was obtained, using the pROC library; the result can be found in Figure 11 and demonstrates that the model can successfully distinguish between positive and negative cases (compared with the other metrics, it may be improved, but it is still a good result).

It is important to remember that the accuracy () function takes a model object and a test set as input and returns the proportion of correct predictions on the test set. The precision () function returns a value between 0 and 1, where 1 indicates a perfect model and 0 indicates a model that is no better than random guessing.

A high precision in a model means that it can accurately predict positive cases, but it doesn't inform us about the number of cases it may have missed. This is where the recall() function comes into play. The recall() function helps measure the proportion of correctly predicted actual positives, indicating how well the model can predict positive cases. By utilizing both precision() and recall() functions, we can gain a more comprehensive understanding of the model's performance compared to relying solely on the accuracy() function.

The F1 score is a metric used to evaluate the effectiveness of a binary classification model. It is derived from the harmonic mean of precision and recall, which is a measure of central tendency that considers outliers more than the arithmetic mean. This makes it a reliable way of assessing the performance of a binary classification model, as it can be influenced by a small number of false positives or false negatives. A high F1 score signifies that the model excels in both precision and recall.

Finally, the AUC () function calculates a value ranging from 0 to 1. If the value is 1, it means the model is perfect, whereas if it's 0.5, the model is no better than random guessing. A higher AUC value implies that the model can effectively differentiate between positive and negative cases.

# Convert the predictions variable to a numeric variable

predictions\_numeric <- as.numeric(predictions)

# Calculate the ROC curve

roc\_curve <- roc(test\_set$Churn, predictions\_numeric)

# Calculate the ROC AUC

roc\_auc <- auc(roc\_curve)

# Print the ROC AUC

print(paste('ROC:', roc\_auc))

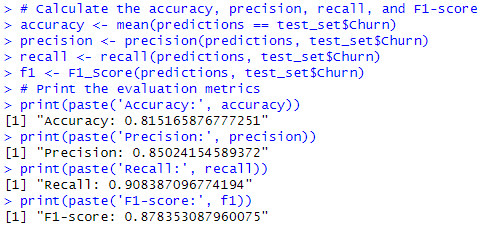


Figure . Results (in console) regarding Accuracy, Precision, Recall and F1-Score of the model.

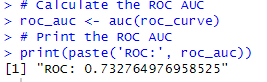
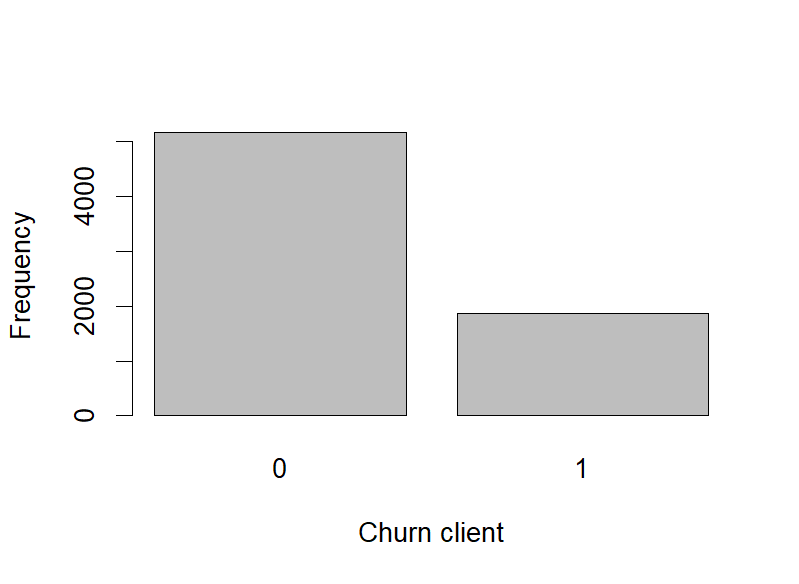


Figure . Result (in console) of the ROC.

1. Addressing Class Imbalance

As we plot the frequency of results of the response variable, we can observe that there is a skewness between the categorical results and “0”. In other words, most of the data have a NO CHURN category.



Graph. Frequency bar plot of Churn incurrence among the clients; 0 represents “NO CHURN”, 1 “YES CHURN”

There are several methods or options for handling imbalanced classes. These are listed approximately in order of effort:

* Do nothing. You can train on the so-called natural (or stratified) distribution; sometimes, it works without modification.
* Balance the training set in some way:
  + Oversample the minority class.
  + Under sample the majority class.
  + Synthesize new minority classes.
* Throw away minority examples and switch to an anomaly detection framework.
* At the algorithm level, or after it:
  + Adjust the class weight (misclassification costs).
  + Adjust the decision threshold.
  + Modify an existing algorithm to be more sensitive to rare classes.
* Construct an entirely new algorithm to perform well on imbalanced data.

For our case, we used the balanced training approach to under sample the majority class (when the client churns or category 0). Down-sampling would randomly sample the first class to be the same size as the second class (so only a percentage of the total training set is used to fit the model). caret contains a function (down Sample) to do this.

Once we implemented the code in R, both categories were balanced; instead of using 3613 and 1309 result observation for category “No” and “Yes” for variable Churn, respectively, the train data set changed towards a dataset where both categories had 1309 values.

The same model was used for unbalanced data and the same testing set was used to evaluate the model with the new train set.

The table below displays the results of comparing unbalanced and balanced data. The scoring tests show similar results, but the balanced data scores are lower in statistics, except for the AUC and Precision scores. This means that the balanced data is capable of predicting “positive” (churn) results, but the other parameters indicate that it misses the other categorical values. Therefore, the unbalanced class dataset might be better for the prediction analysis.

|  |  |  |
| --- | --- | --- |
| Statistic | Unbalanced Class | Balanced Class |
| Accuracy | 0.8151 | 0.7511 |
| Precision | 0.8502 | 0.9190 |
| Recall | 0.9083 | 0.7251 |
| F1-score | 0.8783 | 0.8106 |
| AUC score | 0.7327 | 0.7327 |

Texto

Descripción generada automáticamente

Figure Model results from balanced data.

Texto

Descripción generada automáticamente

Figure Scoring statistics from a balanced model

**3. Conclusions**

In general terms, logistic regression is a useful tool to determine regression models for those datasets that, due to their complex characteristics, cannot be fitted into a linear model. In this document, given a dataset concerning the churn of customers for a certain company (and several variables that may be related to the churn rate), a logistic regression model was fitted and testes. It was discovered that the dataset had several challenges to be analyzed, specifically missing values and different type of data (categorical and numerical); this was dealt cleaning the dataset.

Once the model was developed and tested, it was verified to be a good model (with several evaluation parameters proving the adequacy of the model). The resulting model included just a part of the totality of variables (Senior Citizen, Dependents, Tenure, Multiple Lines, Internet Service, Online Security, Tech Support, Streaming TV, Streaming Movies, Contract, Paperless Billing, Payment Method, Monthly Charges, Total Charges), and, in order for the company to achieve better results, it must focus in the most relevant ones. According to the p-values, the 3 most important variables are: Tenure, Internet Service and Contract; so, they must focus on these variables and then define designs of experiments to understand the effect of changing the variables in the outcome.

It is important to mention that, even if the Tenure is a highly relevant variable, it is difficult to deal with, as it relies mostly on factors external to the company (it may even be studied is a different model as a response variable); so, the main activities that the organization should do must be related to Internet Service and type of Contract. For this, it would be recommended to do a design of experiments to identify the type of services in these 2 variables that lead to better results, and then understand what has been done better in that category to expand it to the rest.

**4. Repository**

[**https://github.com/JoseGerardoHuertaDeRubin/LogisticRegression04oct2023**](https://github.com/JoseGerardoHuertaDeRubin/LogisticRegression04oct2023)

**References**

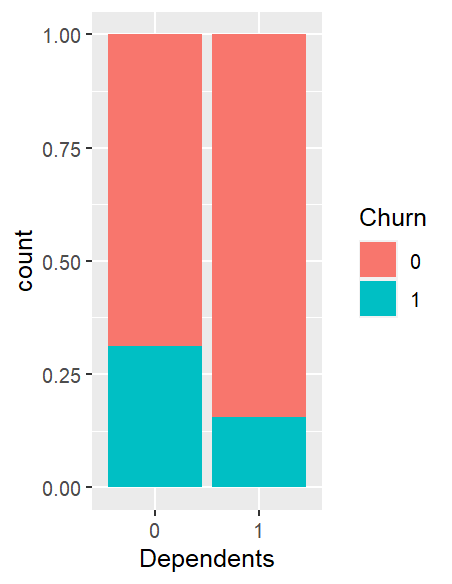
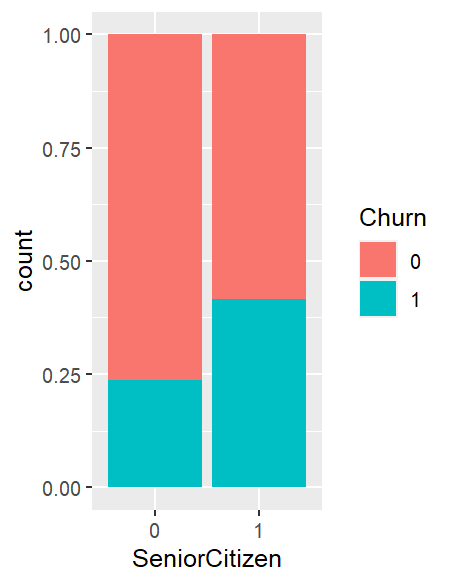
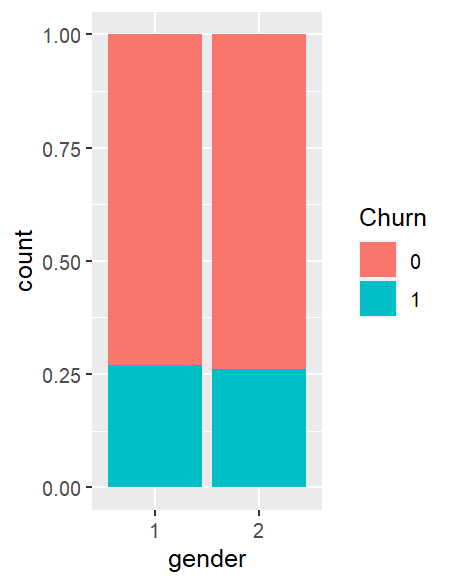
Martin, D. P. (2023, March 8). *Imbalanced classes in R: Part 1. [Blog post*]. Retrieved from <https://dpmartin42.github.io/posts/r/imbalanced-classes-part-1>

Kuhn, M., (n.d.). *11 Subsampling For Class Imbalances*. Github. [11 Subsampling For Class Imbalances | The caret Package (topepo.github.io)](https://topepo.github.io/caret/subsampling-for-class-imbalances.html)

Kuhn, M., (n.d.). *The caret Package*. Github. [The caret Package (topepo.github.io)](https://topepo.github.io/caret/index.html)

**APPENDIX A. VISUALIZATION PLOTS**

In Appendix A, we can visualize bar plots of frequency distributions categorized by the Churn variable of the dataset. If Churn equals zero, the client did not churn; otherwise, they did churn. In the next table is described the categorizing values for the variables plotted.

****

