UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Data science

Machine Learning Model for Telco Churn Company

Integrantes:
Lucas Mareque 1711647

Índice

Introduction and Objectives

This project aims to apply the knowledge acquired from the Data Science course to a real business case in order to reinforce it and provide a first real experience in the field of application. It also seeks to evaluate the effectiveness that can be obtained by assessing different models for this same business case.

The business case to be analyzed is the "Telco Churn" case, with the following requirement:

"Telco NN has asked for your help in predicting which customers will leave the company. You will be provided with a dataset of 7,043 customers, containing 21 variables that show some characteristics of the company's clients."

In summary, the company is looking for a model that will allow them to predict, with some accuracy, which customers are most likely to stop using their services.

Dataset description

The dataset provided by the company is as follows:

Telco churn dataset dictionary			
Variable	**Descripción**	**Tipo de dato**	**Valores posibles**
Customer ID	Customer identifier value	object	
gender	Customer gender	object	Female, Male
SeniorCitizen	Whether the customer is a Senior Citizen or not	float	
Partner	Whether the customer has a partner or not	object	Yes, No
Dependents	Whether the customer has dependents or not	object	Yes, No
tenure	Customer tenure	float	
PhoneService	Whether the customer has a phone service or not	object	Yes, No
MultipleLines	Whether the customer has multiple lines or not	object	Yes, No, No phone Service
InternetService	Type of internet service the customer receives, if any	object	No, DSL, Fiber optic
OnlineSecurity	Whether the customer has online security service or not	object	Yes, No, No internet Service
OnlineBackup	Whether the customer has backup service or not	object	Yes, No, No internet Service
DeviceProtection	Whether the customer has device protection or not	object	Yes, No, No internet Service
TechSupport	Whether the customer has tech support or not	object	Yes, No, No internet Service
Streaming TV	Whether the customer has a streaming service or not	object	Yes, No, No internet Service
Streaming Movies	Whether the customer has a movie streaming service or not	object	Yes, No, No internet Service

Telco churn dataset dictionary			
Variable	**Descripción**	**Tipo de dato**	**Valores posibles**
Contract	Customer contract type	object	Month-to-month , One year, Two year
Paperless Billing	Whether the customer receives a paper bill or not	object	Yes, No
PaymentMethod	Customer payment method	object	Electronic check, Mailed check, Bank transfer (automatic) 'Credit card (automatic)'
MonthlyCharges	Monthly cost	float	
TotalCharges	Total charges	object	
Churn	Whether the customer has left the company or not	object	Yes, No

Tabla 2.1 - Diccionario Telco Churn

The dataset contains 7,042 records, of which only 847 do not have any missing values (NaN).

For data preprocessing, the following hypotheses were considered:

- If a customer's record indicates no internet service, the series of fields dependent on having internet (*OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies*) will be set to "No". Conversely, if any of these fields had the value "No internet Service," the rest of the fields will be set to "No."

- For float variables, NaN values will be replaced by the mean of the column in the dataset to preserve as many records as possible.

- The CustomerID column does not contain relevant information for the analysis and is therefore removed.

After processing, we have a dataframe with 1,276 records and 20 columns, managing to retain 429 additional records that initially contained null values.

For more details, refer to the attached Jupyter Notebook on Pre-Processing.

Exploratory Data Analysis

To begin the exploratory data analysis, we start with the probability distribution of the Churn variable, which is what we aim to predict. As shown in *Figure 3.1*, the probability of customer churn is 74.61%.

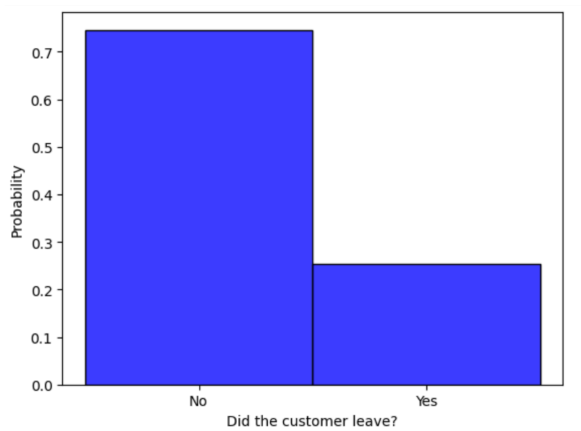


Figure 3.1 - Probability Distribution of Customer Churn[1]

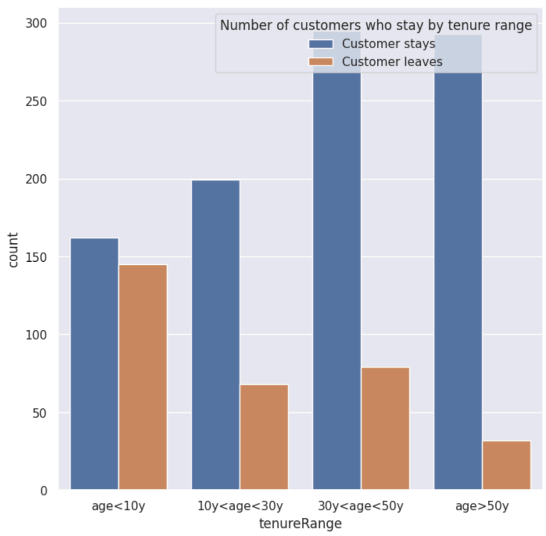


Figure 3.2 - Customer Churn Distribution by Tenure Ranges[2]

After analyzing correlations, we proceed to examine the relationship between tenure and Churn. To do this, tenure ranges are created, and the following relationship is found. In *Figure 3.2*, we observe a decreasing trend in customer churn as tenure increases.

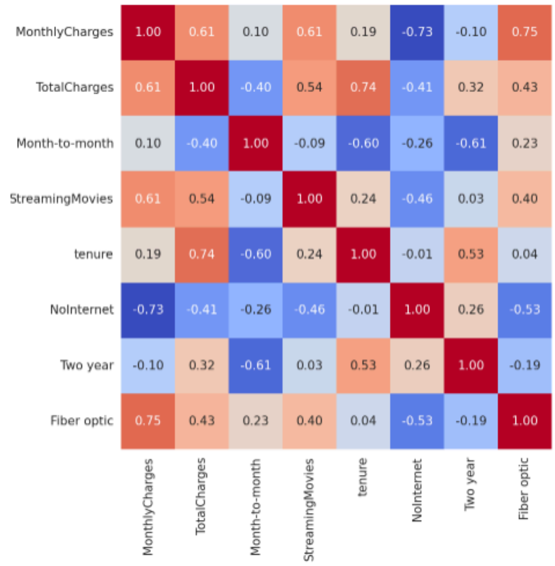


Figure 3.3 - Correlation Matrix of the 8 Most Related Variables[3]

Additionally, correlations between various variables were analyzed, finding that the most related are: MonthlyCharges and Fiber optic (0.75); Tenure and TotalCharges (0.74); MonthlyCharges and TotalCharges (0.61).

For more details, refer to the attached Jupyter Notebook on EDA.

Materials and Methods

The algorithms to be used in the model development are:

- **Logistic Regression**: This is based on traditional regression, which is a statistical method where one variable is explained based on one or more other variables (independent variables). The modification in its logistic version is that the result is binary[4]. It relies on the sigmoid function, which transforms a linear combination of independent variables weighted by coefficients into a value between 0 and 1. The formula for the logistic function is:

$$P(Y = 1) = \frac{1}{1+e^{-(\beta 0+\beta 1X1+\beta 2X2+...+\beta kXk)}}$$

Where P(Y=1)is the probability that the dependent variable equals 1.

β_0 = intercept.

$\beta_1, \beta_2, (...), \beta_k$ = coefficients of the independent variables $X_1, X_2, (...), X_k$

1 Own preparation

2 Own preparation

3 Own Preparation.

4 Hilbe, J. M. 2009. Logistic Regression Model. CRC Press.

- Principal Component Analysis: Principal Component Analysis (PCA) of a data matrix extracts the dominant patterns in the matrix in terms of a complementary set of scores and loading plots[5]. This method involves the following steps: Calculating the covariance matrix of the original data, calculating the eigenvalues and eigenvectors of the covariance matrix. The eigenvectors are ordered according to the eigenvalues in descending order. The original data is then projected onto the space defined by the principal components.

- Neural Network for Classification: Artificial Neural Networks (ANNs) consist of layers of nodes, including an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, is connected to others and has associated weights and thresholds[6]. In classification networks, the key difference is that they allow only one output response for any given input pattern[7]. The main components of a classification neural network are: the structure (with the mentioned layers), the weights and thresholds of connections between nodes, and the learning and backpropagation processes through different epochs (cycles).

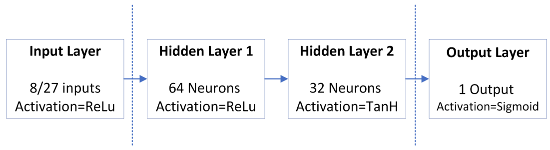
- The model comparison method we will use is comparison of accuracy and AUC-ROC (Area Under the ROC Curve).

The ROC Curve (Receiver Operating Characteristic) is a graphical representation of a classification model's performance across different decision thresholds. The area under this curve (AUC-ROC) measures the model's ability to distinguish between classes. A higher AUC-ROC indicates better performance.
An AUC-ROC of 0.5 suggests performance similar to random guessing, while an AUC-ROC of 1.0 indicates perfect performance.
The ROC curve is created by plotting the true positive rate (sensitivity) against the false positive rate (1 - specificity) at various decision thresholds. A model with a ROC curve closer to the top left corner of the plot has better performance.

Experiments and Results

The procedure used to build the classification model is explained as follows: the dataset used to train the models first undergoes normalization with autoscaling (mean = 0, Standard Deviation = 1). We started with a logistic regression model without parameter tuning (using default parameters from the SciKit Learn library). Then, the same model was then trained with parameter tuning (GridSearch).

Once the results were obtained, the dataset's dimensionality was reduced to 10 components using PCA. The previously developed models were then retrained with this reduced dataset. To compare the results, a third model based on neural networks was also trained to assess how the complexity of the model affected the outcomes, with logistic regression being the simplest and neural networks the most complex. The generated neural network has the following architecture:



The results of the tests for the three models are presented below, both for the original dataset and for the dataset with dimensions reduced by PCA.

Dataset Original			
Results / Parameters	**Logistic Regression**	**Logistic Regression w/GridSearch**	**Neural Network**
Accuracy	80.88%	82.45%	77.74%
AUC ROC	0.8257	0.8348	0.6813
C	1	0.009	n/a
Penalty	L2	L2	n/a

Table 4.1 - Results for Original Dataset

Dataset PCA			
Results / Parameters	**Logistic Regression**	**Logistic Regression w/GridSearch**	**Neural Network**
Accuracy	78.99%	78.99%	80.25%
AUC ROC	0.8271	0.8270	0.8297
C	1	0.3	n/a
Penalty	L2	L2	n/a

Table 4.2 - Results for PCA Processed Dataset

[5] Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. Chemometrics and Intelligent Laboratory Systems, 2(1-3).

[6] IBM. What is a neural network? IBM. [What are Neural Networks? | IBM]

[7] Baughman, D.R., & Liu, Y.A. (1995). Classification: Fault Diagnosis and Feature Categorization.

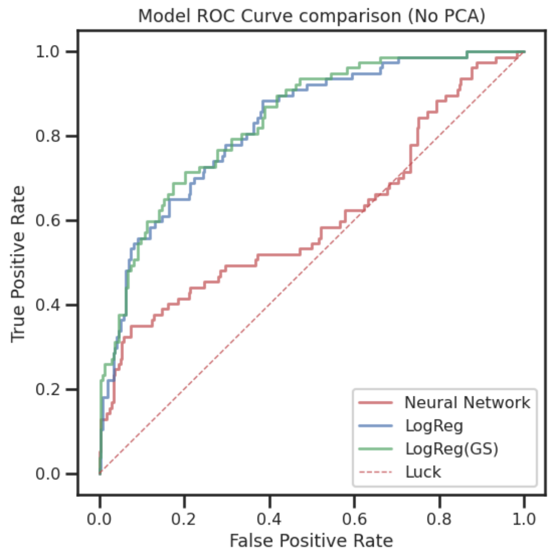


Figure 4.1 - ROC Curve for Models on Unprocessed Data

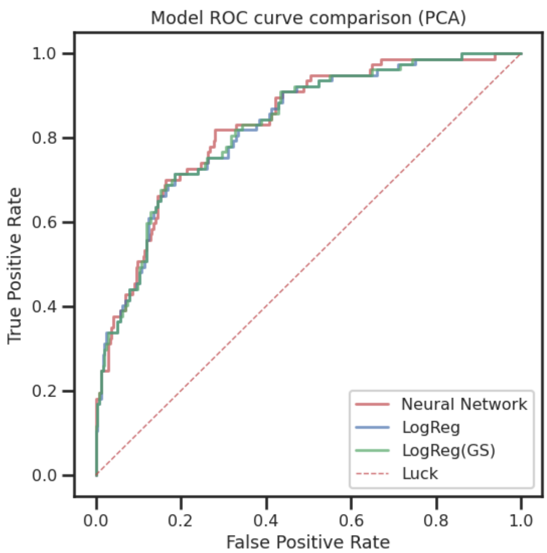


Figure 4.2 - ROC Curve for Models on PCA-Processed Data

For more details, refer to the attached Jupyter Notebook on ML.

Discussion and Conclusions

Based on the results obtained, we can make the following observations and conclusions:

On one hand, the model that best fit the business case was the logistic regression with parameter tuning. It achieved an accuracy of 82.45%, nearly 1.5% higher than its counterpart with default parameters and about 4% better than the neural network. This model is not considered complex, so this indicates that it suggests that the data distribution responds better to a simpler model, avoiding errors due to variance. This indicates that a less complex model, like the parameter-tuned logistic regression, can effectively capture the underlying patterns in the data without overfitting[8].

Continuing along the same line, the results for models using PCA-processed data showed worse performance for both logistic regression models, with the exception of the neural network.

We believe that both effects are due to the same reason mentioned earlier: the complexity of the data does not justify a more complex model.

We do not consider that the model for the available data at the time of analysis can be significantly improved to achieve better accuracy. This is due to the limitation imposed by the high number of missing values in the dataset (resulting in only about 1/7 of the data being usable). With more data, the model could potentially be improved using these same or other classification models.

Finally, the objectives of the development and the report are considered achieved, having gained experience in processing real data, exploring it, and developing various models from scratch, including complex ones like neural networks.

[8] Geman, S., Bienenstock, E., & Doursat, R. (1992). Neural networks and the bias/variance dilemma. Neural Computation, 4(1), 1–58.

References

- Hilbe, J. M. 2009. Logistic Regression Model. CRC Press.
- Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied Logistic Regression. John Wiley & Sons.
- Jolliffe, I. T. (2002). Principal Component Analysis. Springer.
- Baughman, D.R., & Liu, Y.A. (1995). Classification: Fault Diagnosis and Feature Categorization.
- IBM. What is a neural network? IBM. What are Neural Networks? | IBM
- Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep Learning. MIT Press.
- Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. Chemometrics and Intelligent Laboratory Systems, 2(1-3).
- Fawcett, T. (2006). An Introduction to ROC Analysis. Pattern Recognition Letters, 27(8), 861–874.
- Cluster AI 2023 GitHub repository. 2023. clusterai/clusterai_2023
- Geman, S., Bienenstock, E., & Doursat, R. (1992). Neural networks and the bias/variance dilemma. Neural Computation, 4(1), 1–58. https://doi.org/10.1162/neco.1992.4.1.1