UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Ciencia de Datos

Modelo de Machine Learning para Telco-Churn

Integrantes:
Gaspar Rivollier 167609-0

Fecha de entrega:
04/12/2023

Índice

Introducción y Objetivos	**3**
Descripción del Dataset	**3**
Análisis Exploratorio de Datos	**4**
Materiales y métodos	**5**
Experimentos y resultados	**6**
Discusión y conclusiones	**7**
Referencias bibliográficas	**8**

Introducción y Objetivos

Este proyecto tiene por objetivo la aplicación de conocimientos obtenidos del curso de Ciencia de Datos en un caso de negocio real con el fin de reforzarlos y brindar una primera experiencia real en el campo de aplicación. Además busca evaluar la eficacia que se puede obtener evaluando distintos modelos para este mismo caso de negocio.

El caso de negocio que se analizará es el caso "Telco Churn", con el siguiente requerimiento:

"*La Telco NN les pidió que los ayuden a predecir qué clientes dejarán la compañía. Para ellos se les presentará un dataset con una cartera de clientes de 7.043 personas con 21 variables que muestran algunas características de los clientes en la empresa.*"

En resumen, la empresa busca un modelo que les permita predecir con cierta eficacia qué clientes tienen mayor potencial de dejar de usar sus servicios.

Descripción del Dataset

El dataset presentado por la empresa es el siguiente:

Diccionario dataset del Telco churn			
Variable	**Descripción**	**Tipo de dato**	**Valores posibles**
CustomerID	Valor identificador de clientes	object	
gender	Género del cliente	object	Female, Male
SeniorCitizen	Si el cliente es un SeniorCitizien o no	float	
Partner	Si el cliente tiene un socio o no	object	Yes, No
Dependents	Si el cliente tiene dependientes o no	object	Yes, No
tenure	Antigüedad del cliente	float	
PhoneService	Si el cliente tiene un servicio de telefono o no	object	Yes, No
MultipleLines	Si el cliente tiene multiples lineas o no	object	Yes, No, No phone Service
InternetService	Tipo de servicio de internet que recibe. Si es que recibe	object	No, DSL, Fiber optic
OnlineSecurity	Si el cliente tiene un servicio de seguridad online o no	object	Yes, No, No internet Service
OnlineBackup	Si el cliente tiene un servicio de backup o no.	object	Yes, No, No internet Service
DeviceProtection	Si el cliente tiene un seguro del dispositivo o no	object	Yes, No, No internet Service
TechSupp	Si el cliente tiene soporte	object	Yes, No, No
ort	de tecnología o no.		internet Service
Streaming TV	Si el cliente tiene servicio de streaming o no	object	Yes, No, No internet Service
Streaming Movies	Si el cliente tiene servicios de streaming de peliculas o no	object	Yes, No, No internet Service
Contract	Tipo de contrato del cliente	object	Month-to-month, One year, Two year
Paperless Billing	Si el cliente recibe la factura en papel o no.	object	Yes, No
PaymentMethod	Tipo de pago del cliente	object	Electronic check, Mailed check, Bank transfer (automatic) 'Credit card (automatic)'
MonthlyCharges	Costo mensual	float	
TotalCharges	Cargos totales	object	
Churn	Si el cliente se fue de la compañía o no	object	Yes, No

Tabla 2.1 - Diccionario Telco Churn

El dataset cuenta con 7042 registros. De los cuales solo 847 no poseen algún campo nulo (NaN).

Para el preprocesamiento de los datos se tuvieron en cuenta las siguientes hipótesis:

- Si en el registro de un cliente figura que no posee servicio de internet, la serie de campos que dependen de que se tenga internet (*OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies*) serán "No". En sentido inverso, si alguno de estos campos tenía el valor "No internet Service", entonces el resto de campos valdrán "No".
- Para las variables float, los valores NaN serán reemplazados por la media de la columna en el dataset, para conservar la mayor cantidad de registros posibles.
- La columna CustomerID no contiene información relevante para el análisis, por lo que es eliminada.

Finalizado el procesamiento, tenemos un dataframe de 1276 registros y 20 columnas. Logrando conservar 429 registros adicionales que contenían nulos inicialmente.

Para más detalles, remitirse al Jupyter Notebook adjunto respectivo a Pre-Processing.

Análisis Exploratorio de Datos

Para comenzar el análisis exploratorio de datos, partimos de la distribución de probabilidad de la variable Churn, que es la que intentamos predecir. Se observa en la *Figura 3.1* la probabilidad de salida de clientes, esta es de 74.61%

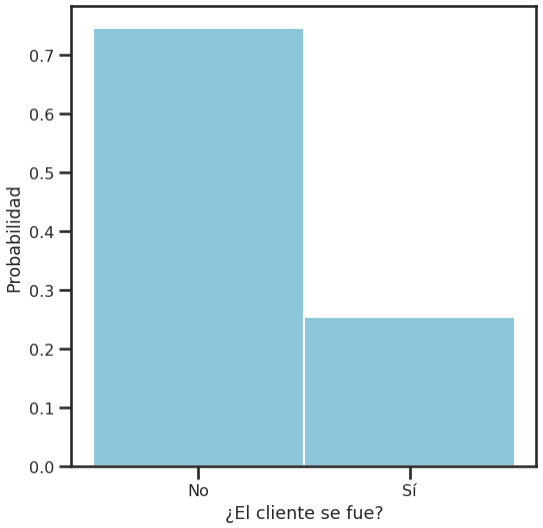


Figura 3.1 - Distribución de probabilidad de salida de clientes[1]

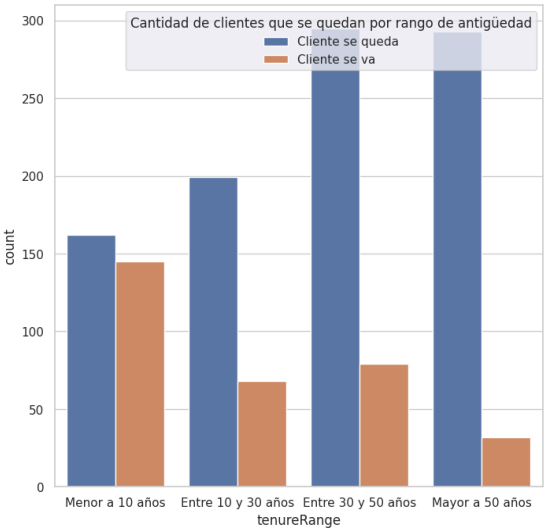


Figura 3.2- Distribución de salida de clientes en rangos de antigüedad[2]

Luego de un análisis de correlaciones, se procede a analizar la relación de tenure con Churn. Por lo que se crean rangos de tenure, y se encuentra la siguiente relación. En la *Figura 3.2* vemos que hay una tendencia decreciente de marcharse de la compañía en la medida que aumenta la antigüedad.

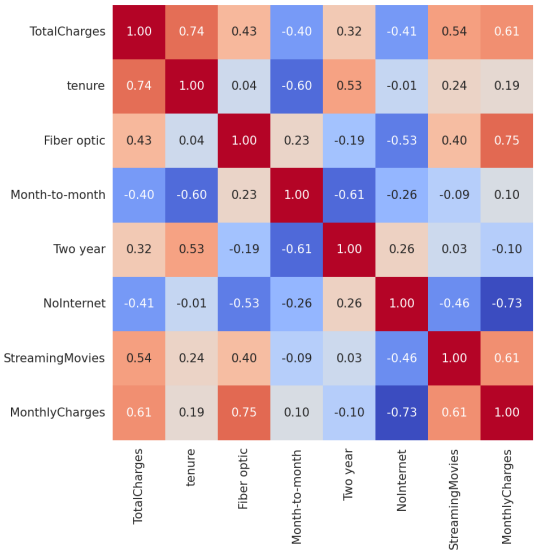


Figura 3.3 - Matriz de correlación de 8 variables más relacionadas[3]

Además, se analizaron las correlaciones entre las distintas variables, encontrando que las más relacionadas son: MonthlyCharges y Fiber optic (0.75); Tenure y TotalCharges (0.74); MonthlyCharges y TotalCharges(0.61)

Para más detalles, remitirse al Jupyter Notebook adjunto respectivo a EDA.

Materiales y métodos

Los algoritmos a utilizar en el desarrollo del modelo son:

- Regresión logística: Parte de la regresión tradicional, que es un método estadístico donde una variable es explicada en base a otra u otras variables (variables independientes). La modificación en su versión logistica, es que el resultado es binario[4]. Se basa en la función sigmoide, la cual transforma una combinación lineal de variables independientes ponderadas por coeficientes en un valor entre 0 y 1. La fórmula para la función logística es:

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

Siendo P(Y=1) probabilidad de que variable dependiente sea = 1.

[1] Elaboración propia.
[2] Elaboración propia.
[3] Elaboración propia.
[4] Hilbe, J. M. 2009. Logistic Regression Model. CRC Press.

β_0 = ordenada al origen.

β_1, β_2, (...) , β_k = coeficientes de variables independientes X_1, X_2, (...) , X_k

- Principal Component Analysis: El análisis de componentes principales de una matriz de datos extrae los patrones dominantes en la matriz en términos de un conjunto complementario de gráficos de puntuación y de carga[5]. Este método consiste en los siguientes pasos: Cálculo de la matriz de covarianza de los datos originales, Cálculo de autovalores y autovectores de la matriz de covarianza. Se ordenan los autovectores según los autovalores en orden descendente. Los datos originales se proyectan sobre el espacio definido por las componentes principales.

- Neural Network para clasificación: Las redes neuronales artificiales (RNAs) están compuestas por capas de nodos, que incluyen una capa de entrada, una o más capas ocultas y una capa de salida. Cada nodo, o neurona artificial, se conecta con otro y tiene un peso y umbral asociados[6]. La diferencia con las de clasificación es que estas solo permiten una respuesta de salida para cualquier patrón de entrada[7]. Las componentes principales de una red neuronal de clasificación son: La estructura (Con las capas mencionadas), los pesos y umbrales de las conexiones entre nodos, y el aprendizaje y retropropagación en los distintos epochs (ciclos).

- El método de comparación entre modelos que utilizaremos es comparación de accuracy y AUC ROC (Área bajo curva ROC).

La curva ROC (Receiver Operating Characteristic) es una representación gráfica del rendimiento de un modelo de clasificación en diferentes umbrales de decisión. El área bajo esta curva (AUC-ROC) mide la capacidad del modelo para distinguir entre clases. Una AUC-ROC más alta indica un mejor rendimiento.

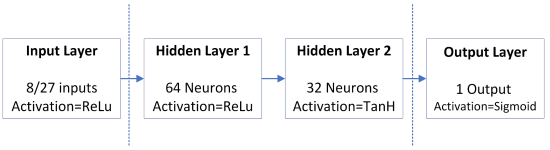
Una AUC-ROC de 0.5 sugiere un rendimiento similar al azar.Una AUC-ROC de 1.0 indica un rendimiento perfecto.

La curva ROC se crea al trazar la tasa de verdaderos positivos (sensibilidad) frente a la tasa de falsos positivos (1 - especificidad) en varios umbrales de decisión. Un modelo con una curva ROC que se acerca más al rincón superior izquierdo del gráfico tiene un mejor rendimiento.

Experimentos y resultados

Se explica a continuación el procedimiento utilizado para realizar el modelo de clasificación: La base de datos utilizada para entrenar los modelos, primero pasan por una normalización con autoscaling (media = 0, Desvío Std. = 1). Se comenzó utilizando un modelo de regresión lineal logística sin búsqueda de parámetros (utilizando predeterminados de librería SciKit Learn). Luego, se entrenó este mismo modelo con búsqueda de parámetros (GridSearch).

Una vez hallados los resultados, se redujo la dimensionalidad del dataset a 10 componentes, mediante PCA. Y se volvieron a utilizar los modelos desarrollados previamente. Para comparar los resultados obtenidos, se entrenó además un tercer modelo basado en redes neuronales para evaluar cómo afectaba en los resultados la complejidad del modelo utilizado, siendo el de regresión lineal el más sencillo y el de redes el más complejo. La red neuronal generada presenta la siguiente arquitectura:



A continuación se presentan los resultados de las pruebas de los 3 modelos, tanto para el dataset original, como para el dataset con dimensiones reducidas por 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

Tabla 4.1 - Resultados para Dataset Original

[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.

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

Tabla 4.2 - Resultados para Dataset procesado con PCA

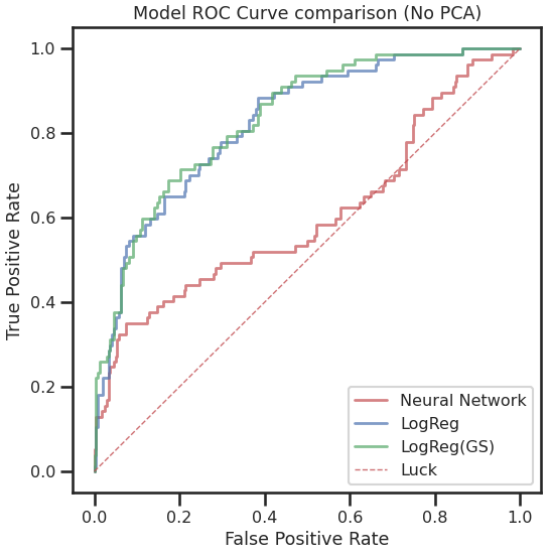


Figura 4.1 - Curva ROC para modelos en datos sin procesamiento PCA

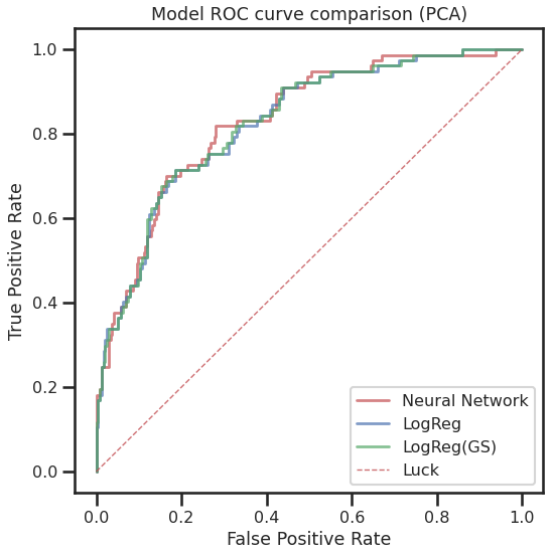


Figura 4.2 - Curva ROC para modelos en datos procesados con PCA

Para más detalles, remitirse al Jupyter Notebook adjunto respectivo a ML.

Discusión y conclusiones

Con los resultados obtenidos, podemos notar las siguientes observaciones y conclusiones:

Por un lado, el modelo que mejor se ajustó al caso de negocio fue la regresión lineal logística con búsqueda de parámetros. Logrando una eficacia de 82.45%, casi 1.5% superior a su contraparte con parámetros predeterminados, y alrededor de 4% mejor que la red neuronal. Este modelo no se considera complejo, por lo que esto habla de que la distribución de los datos responde mejor a un modelo más sencillo, evitando generar error por Variance[8].

Siguiendo esta misma línea, los resultados de los modelos con los datos previamente procesados con PCA, arrojaron en ambas regresiones logísticas peores resultados, siendo la excepción la red neuronal.

Consideramos que ambos efectos responden a la misma razón mencionada previamente, la complejidad de los datos con los que se trabaja, no es tal para que justifique un modelo de tal complejidad.

No consideramos que el modelo para los datos disponibles al momento del análisis, se pueda mejorar mucho más para lograr mejor eficacia. La razón de esto siendo que la cantidad de datos que se pueden utilizar de la base es limitada por la gran cantidad de nulos presentes (resultando en solo 1/7 de la base utilizable, aproximadamente). Con más datos, se podría potencialmente mejorar el modelo utilizando estos mismos u otros modelos de clasificación.

Finalmente, el objetivo del desarrollo y el informe se considera alcanzado, teniendo la experiencia de procesar datos de bases reales, explorar los datos y generar distintos modelos desde cero, incluso complejos como redes neuronales.

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

Referencias bibliográficas

- 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