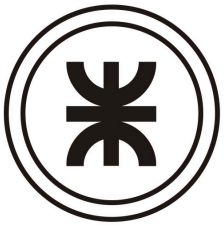


UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

CIENCIA DE DATOS

Trabajo final grupal: Telco Churn

Alumnos:

- Saavedra Parra, Jaider Stiven
- Schiliro, Agustín

Introducción y objetivos

La empresa Telco NN solicitó asistencia técnica para predecir los clientes que dejarán la compañía. Para ello, presentaron un dataset de clientes compuesto por 7.043 clientes y diferentes variables. En las secciones posteriores se describe el dataset seguido de un análisis exploratorio de los datos, para luego alimentar un modelo de Machine Learning y evaluar los resultados correspondientes.

Descripción del dataset

El dataset se compone de 7.043 samples y 21 features. Entre los diferentes atributos, algunos que se pueden identificar son la antigüedad del cliente, el uso de líneas telefónicas, la suscripción a servicios de streaming, el tipo de contrato y de pago y los cargos totales, entre otros. La label del dataset es la variable “Churn”, la cual describe si el cliente se fue de la compañía o no.

Los dtypes presentes en el dataset son 3: int64, object y float. La mayoría de las variables son categóricas, donde se pueden observar un grupo de binarias, ya sea por Yes/No (Partner, Dependents, PaperlessBilling) o por Male/Female (gender). El otro gran grupo puede adoptar hasta 3 valores, donde se encuentran variables como PhoneService, Contract y PaymentMethod. Por último están las variables numéricas, en donde se identifican SeniorCitizen, Tenure y MonthlyCharges.

Lo primero que se identifica es la codificación incorrecta de dos variables: SeniorCitizen y TotalCharges. La primera, que devuelve valores “0.0” y “1.0” es erróneamente un float64. Al tratarse de valores binarios, corresponde utilizar int64. TotalCharges por su parte es presentada como una variable de tipo object, por lo que es necesario transformarla en float64 para su posterior análisis.

Seguido de esto, se observan dos variables que pueden dificultar la operación del modelo: Unnamed: 0 y customerID. La primera describe la numeración de la serie, por lo que optamos por eliminarla. customerID tomará su lugar al convertirla en índice.

Al contar la cantidad de valores nulos, pudimos encontrar 13.959 celdas vacías. Si hubiésemos eliminado los samples con valores null, el dataset hubiese quedado con 844 samples, por lo que optamos por explorar diferentes formas de llenarlo.

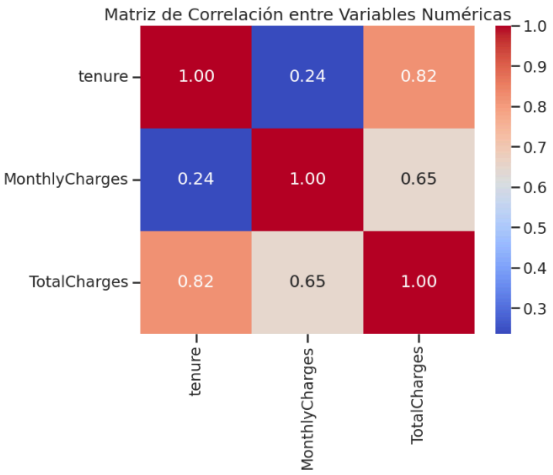
Análisis exploratorio de datos

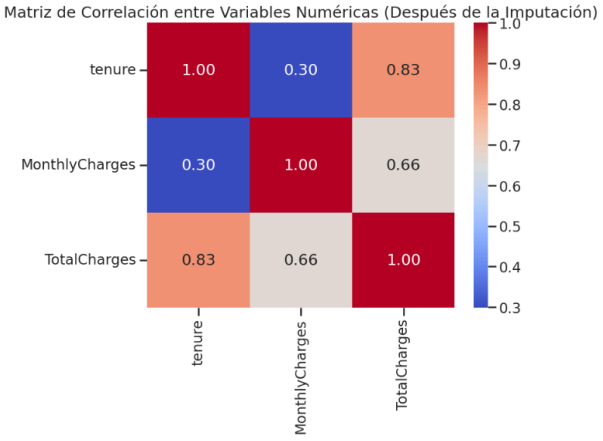
Estudiamos el dataset en búsqueda de patrones para llenar los espacios vacíos. Lo primero que observamos fue que siempre que alguno de los valores de OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV y StreamingMovies eran iguales a “No internet service”, todos los demás tomaban el mismo valor. Por ello, aplicamos una función para que verifique las celdas vecinas, y en caso de encontrar dicho valor, lo llenaba en el espacio vacío. Esto nos permitió llenar 372 celdas. Además, notamos que esta

condición llevaba a que la variable InternetService sea “No” para estos casos, por lo que logramos llenar otras 183 celdas.

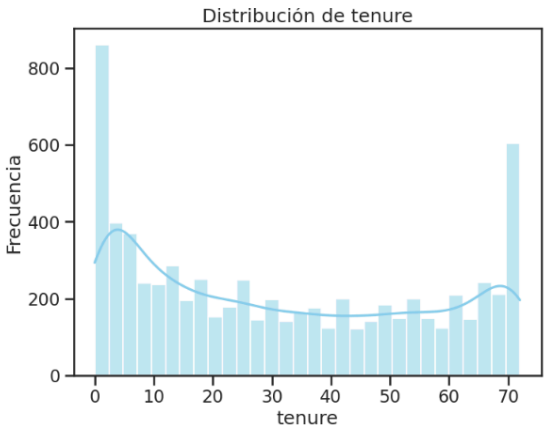
Encontramos otro patrón con las variables PhoneService y MultipleLines. Si MultipleLines es igual a “No phone service”, por obligación PhoneService debía ser “No”. De la misma manera, si encontrábamos un “No” o “Yes” en MultipleLines significa que PhoneService es igual a “Yes”, logrando llenar otras 906 celdas.

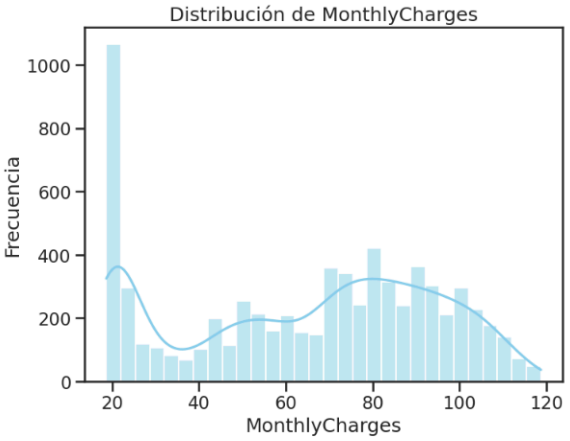
Observamos también que había una relación entre tenure, MonthlyCharges y TotalCharges. Nos dimos cuenta de que tenure * MonthlyCharges devolvía un número con un error del 5% del valor de TotalCharges, por lo que evaluamos su correlación a través de un heatmap. En efecto, tenure y TotalCharges tenían una correlación de 0.82, lo que nos llevó a utilizar un imputador de tipo KNN para llenar las celdas nulas. Posteriormente, volvimos a aplicar el heatmap y encontramos que la correlación era similar, por lo cual conservamos el método.





Luego utilizamos boxplots e histogramas para identificar outliers en las variables numéricas. No encontramos outliers significativos, pero a través de los histogramas pudimos sacar algunas conclusiones: tenure tiene picos de usuarios nuevos y antiguos, mientras que MonthlyCharges tiene un pico al principio. Esto puede asociarse a que la mayoría de los usuarios utilizan suscripciones básicas ($20).

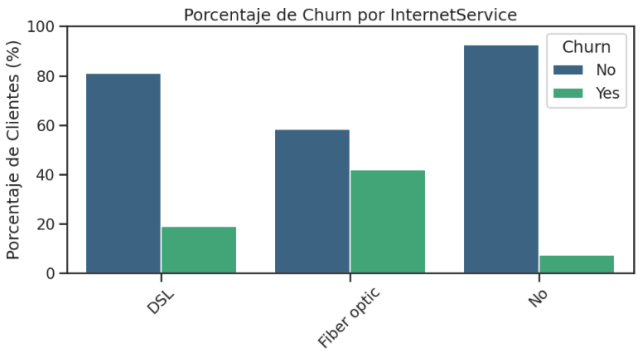


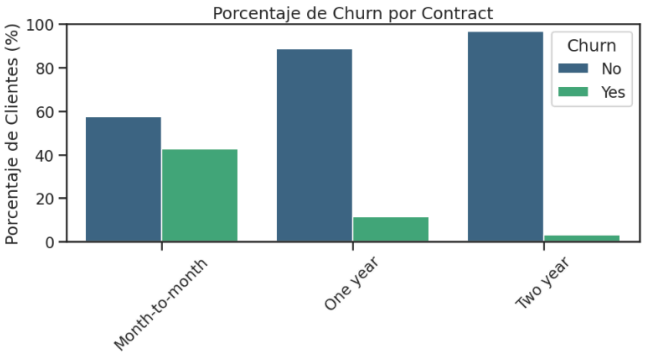


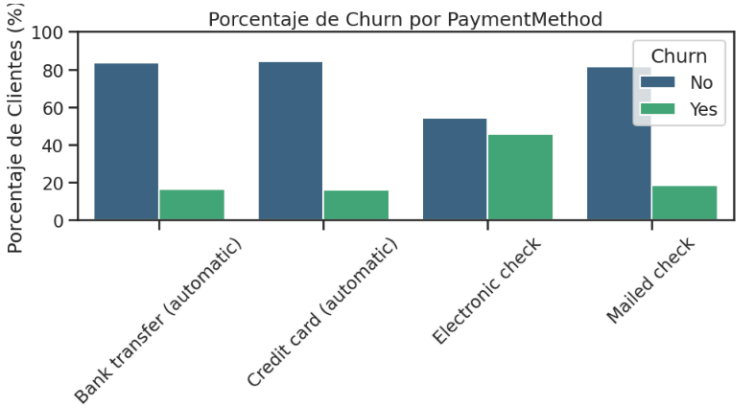
Para llenar las variables categóricas restantes, previo a realizar el llenado de las celdas vacías analizamos a través de gráficos de barras la incidencia de las distintas variables sobre los resultados de Churn. Observamos que la diferencia en las tasas para las variables gender, PhoneService y MultipleLines es mínima, por lo que consideramos que no son relevantes para el modelo y las eliminamos.

Notamos similitudes importantes en distintos grupos de variables: tanto Partner y Dependents como StreamingTV y StreamingMovies muestran distribuciones parecidas, por lo que conservamos una sola por par. Además, el grupo de OnlineSecurity - OnlineBackup - DeviceProtection - TechSupport mostraba lo mismo, por lo que nos quedamos con OnlineSecurity.

Entre otras cosas, observamos mayor tasa de deserción para los usuarios de fibra óptica. Notamos también que los usuarios que no tienen contratado OnlineSecurity tienen mayor probabilidad de abandonar el servicio. Además, vimos que los usuarios que pagan mes a mes tienen una tasa de deserción mayor a los que contratan por 1 o 2 años. Finalmente, entre los métodos de pago el cheque electrónico fue el de mayor abandono respecto de sus pares.





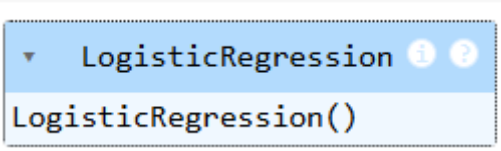


Procedemos entonces a eliminar las variables categóricas no relevantes. Posterior a esto, estudiamos la cantidad de samples que quedarían en el modelo según la cantidad de features vacíos por sample. Procedimos a eliminar todos los registros que tenían 4 o más celdas vacías, resultando en 6.988 samples. Llenamos las celdas vacías en las variables categóricas imputando por moda, y finalmente generamos las respectivas dummies para terminar de preparar el dataset.

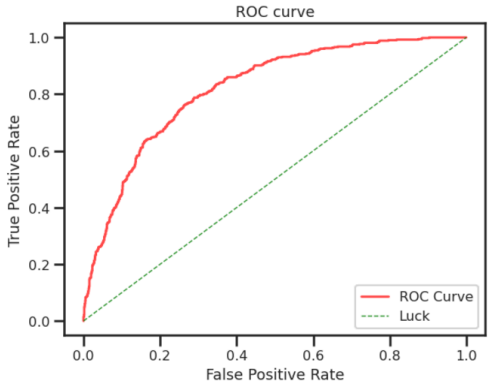
Métodos

Partimos del tipo de problema que tenemos, es un problema de clasificación por variables categóricas. En base a eso, importamos la librería de **SKLEARN** y procedimos a generar un modelo de Aprendizaje Supervisado por Clasificación, el método elegido fue **Logistic Regression**

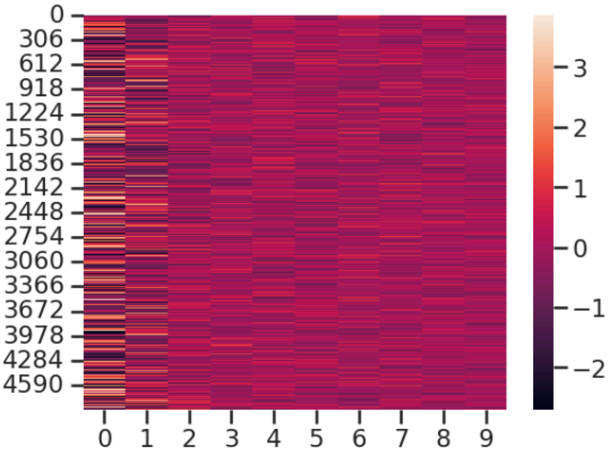
1. Modelo de entrenamiento de xtrain, ytrain, xtest, ytest (70% para Train)
2. model_lr.fit(xtrain_scal, ytrain)



3. Cálculo del AUC ROC



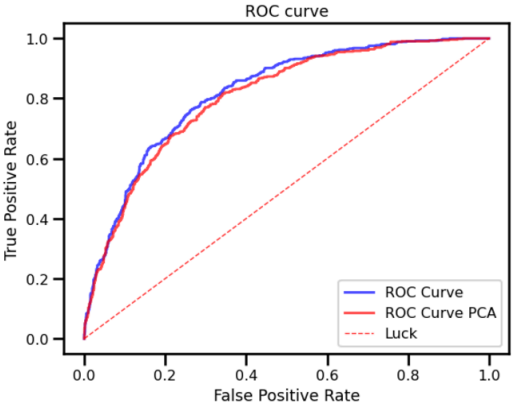
Por otro lado, usamos el método de reducción de dimensionalidad de PCA con 10 autovectores y autovalores.



Resultados

Con el método de **Logistic Regression** llegamos a un AUC de 0,82 y un Accuracy de 0,79, lo que nos dice que nuestro modelo entra en la categoría de Buena Predicción. Sin embargo, al usar el método de reducción de Dimensionalidad de **PCA** obtenemos unos valores muy similares a los anteriores para nuestra predicción (AUC_PCA de 0,81 y un

Accuracy_PCA de 0,79) pero con 5 features. Esto indica que nuestro modelo tiene una alta predicción a pesar de disminuir sus variables.



Discusión

La mayor parte del tiempo fue empleado en la limpieza y preparación del dataset. Según nuestro análisis, entendemos que podemos prescindir de diferentes variables en próximas extracciones. El modelo no necesitaba tantos features como pensábamos ya que pudimos llegar a una AUC aceptable con los 5 más representativos.

Conclusiones

Obtuvimos un modelo con buen nivel de predicción (0,82) que nos permite clasificar a los posibles clientes que dejarán el servicio, con un error cuadrático medio aceptable. Podemos implementar este modelo para la predicción de clientes que dejarán nuestro servicio.