Pérez Maturana, Cristian

Schettino, Matías Agustín

Reporte técnico - Clasificación por Churn para Telco NN

Introducción y objetivos

El objetivo del presente informe es presentar modelos que puedan predecir con la mayor certeza posible si los clientes de Telco NN van a dejar la compañía o no, en base a una cartera de 7043 clientes y 22 características listadas a continuación y que conforman el dataset:

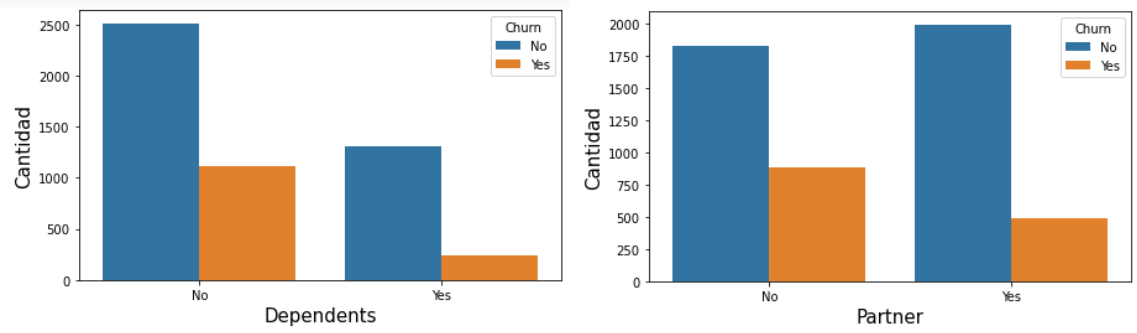
Customer ID, gender, SeniorCitizien, Partner, Dependents, tenure, PhoneService, MultipleLines, InternetService, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies, Contract, PaperlessBilling, PaymentMethod, MonthlyCharges, TotalCharges.

Pre-procesamiento de datos y Exploratory Data Analysis

En primer lugar se nota la presencia de nulos en 14 de las 22 variables, que si son eliminados por su fila completa teniendo al menos un nulo en cualquiera de sus variables, el dataset se reduce de 7043 a 1200, con lo cual para reducir lo menos posible se procede a aplicar criterios que permitan establecer relaciones condicionales entre variables e inferir sus valores nulos:

- **MultipleLines y PhoneService**: variables relacionadas, si MultipleLines indica “No phone service” se rellena nulo con No en PhoneService, y si PhoneService indica “No”, se llena nulo con No en MultipleLines
- **PaperlessBilling y PaymentMethod**: variables relacionadas por clase “Mailed check” presente en PaymentMethod. Si en un nulo de PaperlessBilling se encuentra Mailed check, se imputa “No” en dicha variable, y si se encuentra cualquiera de las otras clases (de pago electrónico), se imputa “Yes”.
- **‘InternetService’ relacionada con las variables 'OnlineSecurity', OnlineBackup','DeviceProtection', 'TechSupport', 'StreamingTV' y 'StreamingMovies’**: si en alguna de estas ultimas variables se encuentra “No internet service” se rellena tanto en InternetService “No” como en el resto de las variables vacías.

A su vez se rellenan nulos en las variables numéricas Monthly Charges y Tenure por sus medias estadísticas, y se verifica que las variables Dependents y Partner guardan una distribución similar entre Churn Yes/No, y proporcionalmente cuentan con más ‘No’ que ‘Yes’ para absorber una imputación arbitraria de valores en estas variables (hacia el ‘No’).

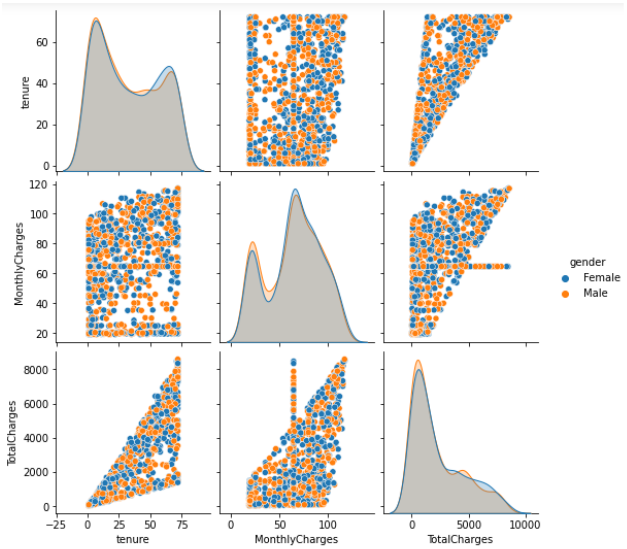


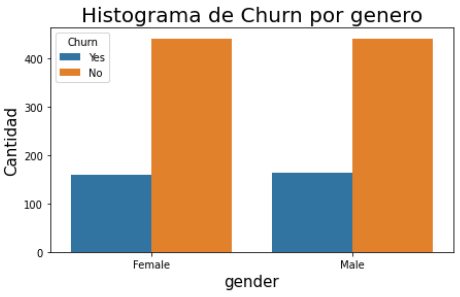
No así con el caso de las variables SeniorCitizen y Contract que al resultar relevantes y difíciles de suponer un criterio de inferencia para ellas, se eliminan las muestras con nulos en su interior. También de aún permanecer nulos en PaymentMethod luego de la primer imputación, se eliminan estas muestras, al igual que los 11 nulls en la variable TotalCharges identificados luego de pasar su formato de string a float. También se limpian duplicados de haberlos para las variables 'customerID' y el índice 'Unnamed', para luego ser sustraídas del dataset por no aportar información al modelo. La variable 'gender' también es extraída para evitar sesgos de género, y luego de estas transformaciones se terminan de unificar los "No internet service" y "No phone service" solamente por 'No', y se proceden a borrar los nulos restantes que no se pudieron imputar, quedando conformado el dataset luego del preprocesamiento por 2484 muestras, que reciben por último un auto-escalado para las variables numéricas, se generan dummies para las categóricas, y se separa el dataset en X e Y (etiquetas Churn) para luego aplicar un train_test_split con un 75-25% de relación entre train y test respectivamente.
Para este manejo de datos se han aplicado las librerías de código abierto en python:
Pandas y Numpy (computación numérica y preprocesamiento de datos), y matplotlib y seaborn para la creación de los gráficos.

<u>Del Análisis exploratorio de datos se verifica lo siguiente:</u>

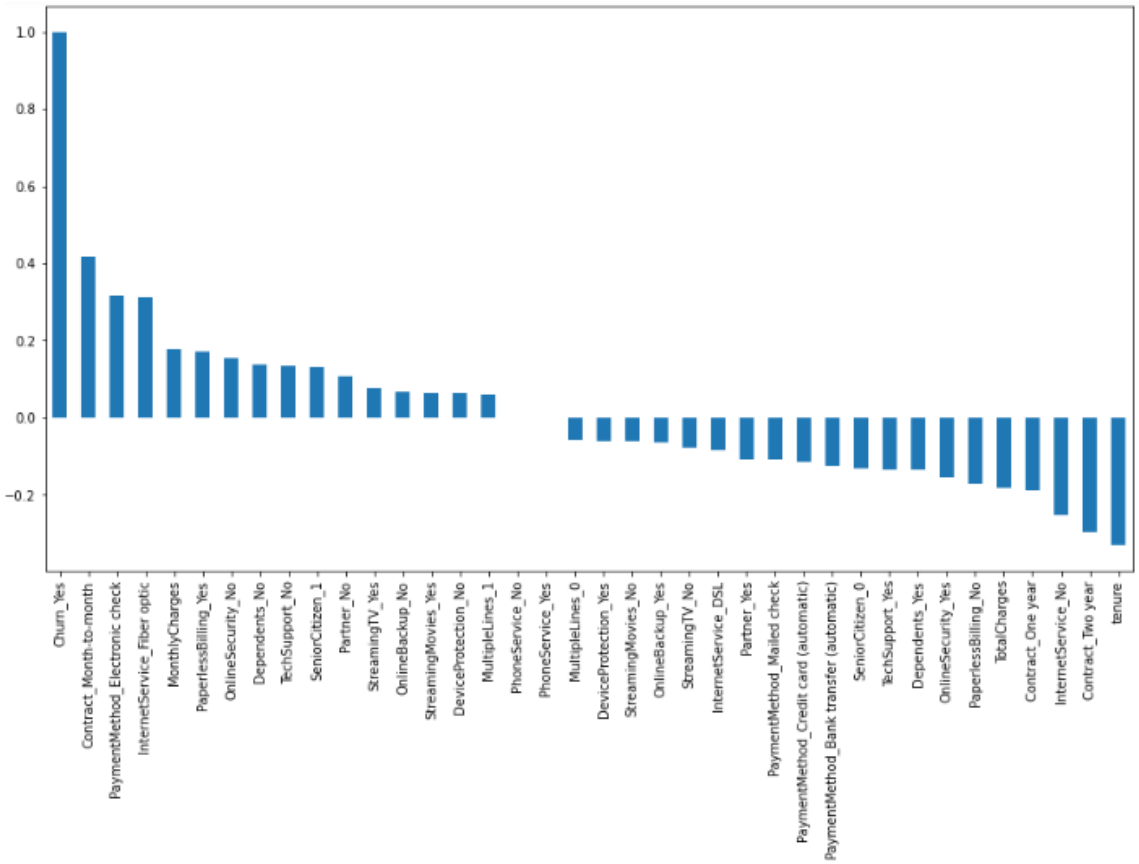
1) Género de los clientes:

No hay una relación directa entre el género de los clientes y ninguna variable, siguen distribuciones similares tanto Hombres como mujeres, y como tienen nulos presentes y para no caer en un sesgo de género para el modelo, se decide eliminar variable del dataset.



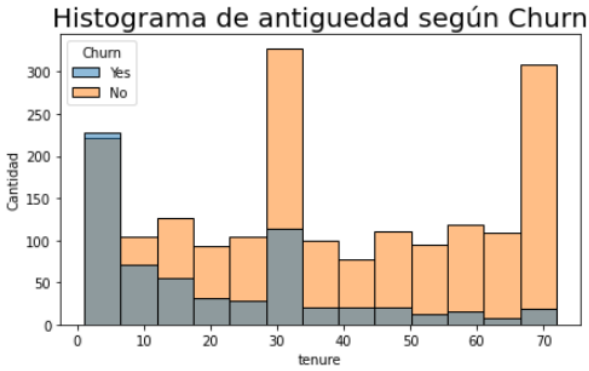


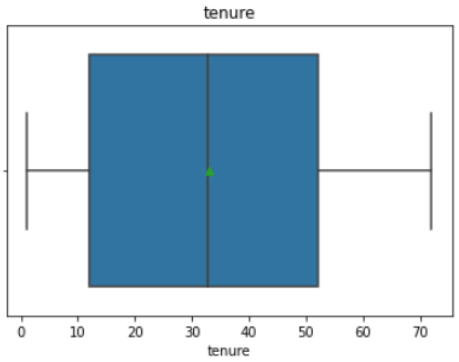
2) Correlación a Churn por variable:



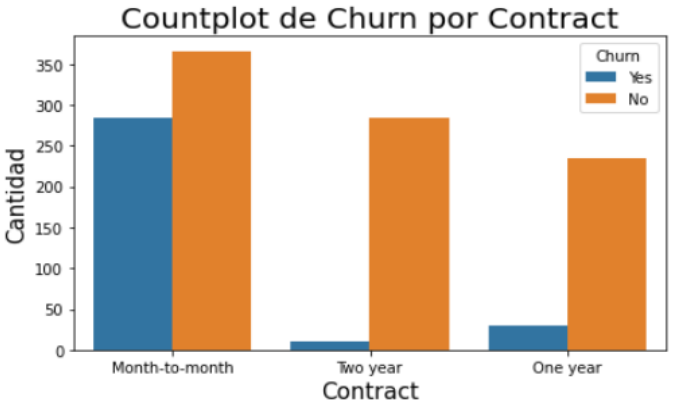
donde se puede observar que **las variables más directamente relacionadas a Churn_yes** son 'Contract_Month-to-month', 'PaymentMethod_Electronic_check' y 'InternetSevice_Fiber optic', mientras que las **más inversamente proporcionales** resultan la antigüedad (clientes más antiguos menos propensos a dejar la empresa), 'Contract_Two year', 'No internet service' y 'Contract_One year'

De esto se puede destacar que casi la mitad de los clientes con antigüedad menor a 7 meses suelen dejar la compañía, y que el 50% de los clientes se encuentra con una antigüedad entre 10 y 55 meses y una mediana de 33.





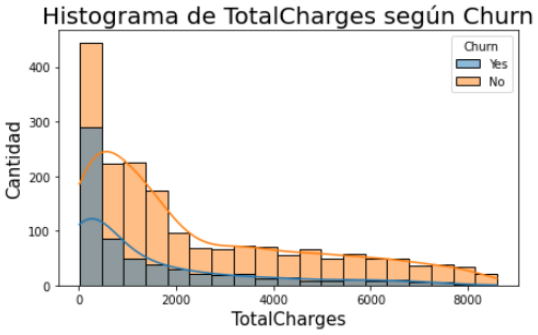
También analizando el tipo de contrato, el más predilecto resulta el de mes a mes, pero al mismo tiempo es el tipo de contrato en el que más clientes prescinden del servicio.

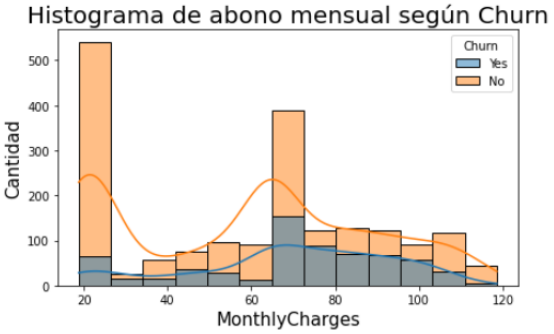


3) Cargos:

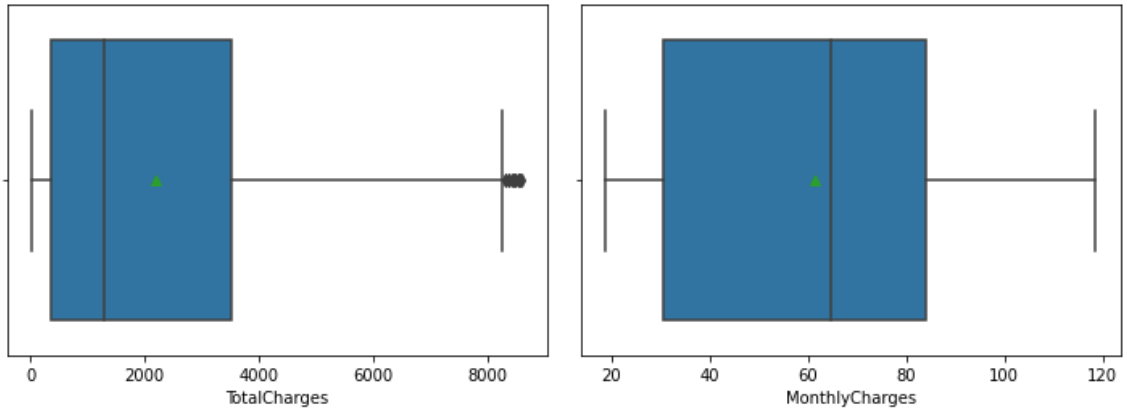
Los pagos mensuales más frecuentes se encuentran en dos rangos: entre 20 y 30, y entre 65 y 75, y también se puede observar que la mayoría de clientes que dejan la firma se encuentran pagando mensualmente más de 100 por mes, mientras que los que pagan menos de 100 por mes raramente dejan la compañía.

En cuanto a los pagos totales la mayor cantidad de clientes se encuentra entre 0 y 1000 de los cuales se hayan distribuciones similares según Churn yes/no.



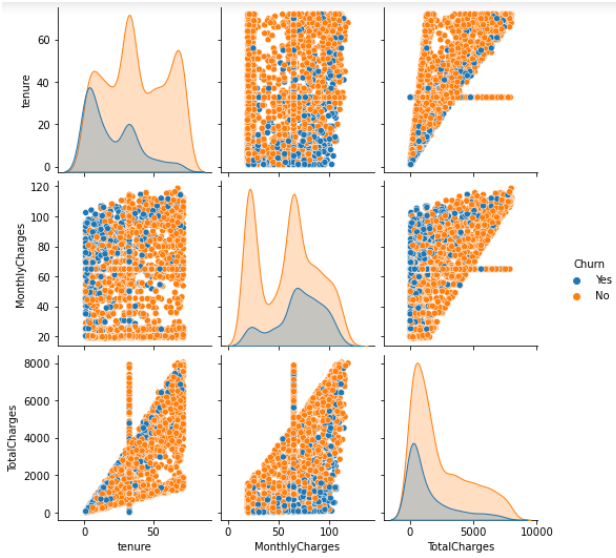


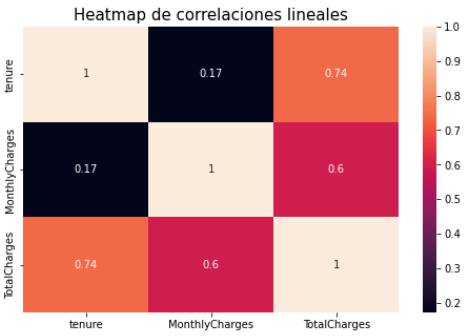
y el 50 % de los clientes gastan entre 350 y 3500 anuales, y entre 30 y 85 mensuales.



(los outliers en TotalCharges por encima del cuantil 0.99 y por debajo del 0.01 son eliminados)

4) Heatmap de correlación de Pearson para variables numéricas, y segmentación por Churn:





en donde se puede observar cierta correlación lineal (0.74) entre antigüedad y TotalCharges.

Como conclusión se puede decir que:

- la antigüedad de los clientes es un factor clave sobre si dejan o no la empresa (a mayor antigüedad es menor la tendencia a dejarla)
- Clientes sin internet son más fieles a mantener el servicio, mientras que los que contratan Fibra óptica son más propensos a dar de baja.
- a mayores gastos mensuales, más clientes dejan la empresa
- el tipo de contrato mes a mes es el que más eligen los clientes que se van, mientras que los contratos a 1 o 2 años retienen a los mismos al menos durante el periodo.

Desarrollo de los métodos de predicción

Para el problema de clasificación presentado, se utilizarán los siguientes modelos de aprendizaje supervisado para predecir la variable Churn:

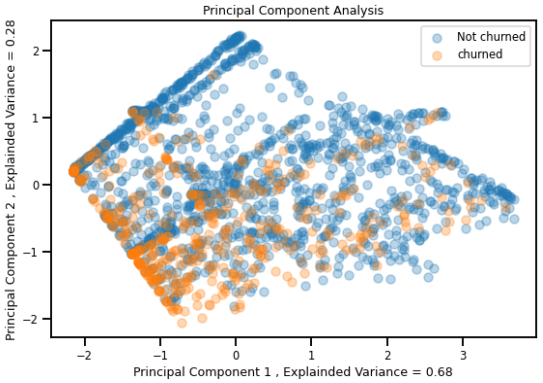
- Support Vector Machine (classificator)
- Regresión logística
- Red neuronal

PCA

Además, se utilizara el método Principal Component Analysis para realizar una reducción de la dimensionalidad para las 3 variables numéricas del dataset (MonthlyCharges, TotalCharges, y tenure) debido a que el PCA no resulta válido para variables del tipo string.

El análisis de componentes principales es un método rápido y flexible, no supervisado, para reducir la dimensionalidad de los datos. [...] El uso del PCA para la reducción de la dimensionalidad implica la eliminación de uno o varios de los componentes principales más pequeños, lo que da lugar a una proyección de los datos en una dimensión inferior que

preserva la varianza máxima de los datos. (Python Data Science Handbook, 2016 VanderPlas). A partir de esta aplicación se logra representar un 95.3% de la variabilidad de los datos en las dos componentes extraídas.



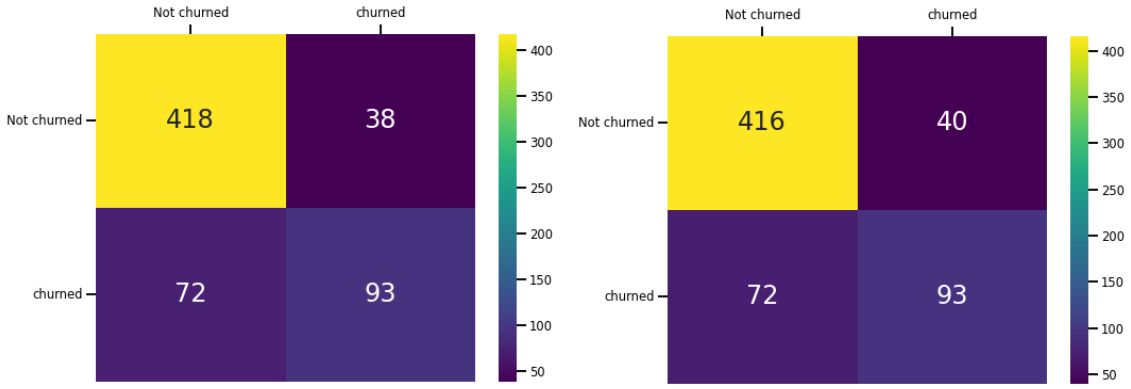
Support Vector Machine (Classificator)

Support vector machine (SVM) es un algoritmo de aprendizaje supervisado que se utiliza en problemas de clasificación y regresión. El objetivo del algoritmo SVM-C es encontrar un hiperplano que separe y maximice el margen entre distintas clases, partiendo de un modelo lineal o bien utilizando non-linear kernels para obtener un hiperplano separador no lineal. En la práctica se suele utilizar junto a un Grid Search, método para buscar los mejores hiperparámetros de un modelo (penalizadores para evitar el over-fitting, o bien distintos kernels posibles), y junto a un Cross-Validation para partir el dataset en K-folds y probar las performances del modelo sobre distintos conjuntos de datos del mismo dataset.

- SVM con kernel lineal

Se define un GridSearch-CrossValidation con kernel linear, cinco folds, y seis costos ‘C’ entre 0.001 y 1000 equidistantes de a un orden de magnitud, para el cual el mejor resultado surge de un C = 1, con un Accuracy de 0.80. A continuación se entrena un modelo idéntico pero al que se le ingresa el dataframe surgido del PCA, para el cual el mejor Costo también resulta 1, y comparten un Accuracy de 0.80

En la matriz de confusión de la izquierda se observan los resultados sin PCA, y en la de la derecha con PCA. El eje vertical de este tipo de gráfico representa las variables reales, y el horizontal las predicciones, resultando útil para ver los tipos de error generados en el modelo, las veces que se predijo bien y cuántas mal según el tipo de variable, encontrándose en este caso valores muy similares para ambos resultados.

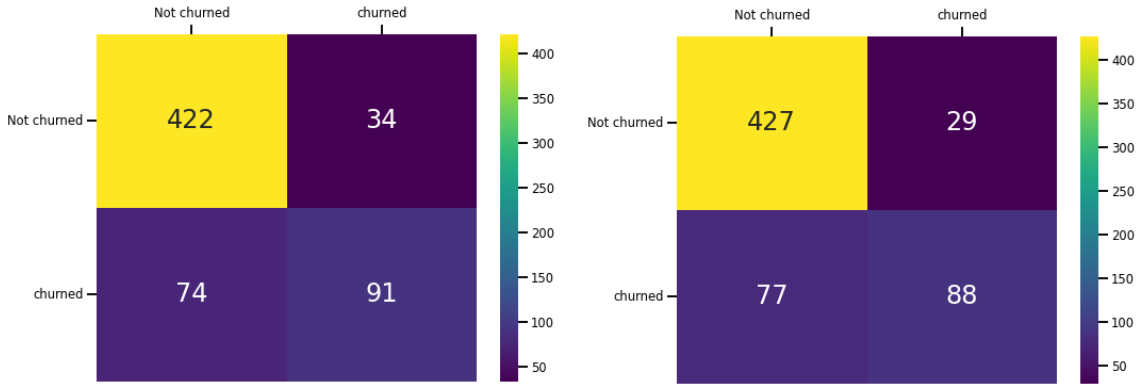


- SVM con kernels no lineales

Se define otro GridSearch-CrossValidation con kernels no lineales (gaussiano y sigmoide), cinco folds, y los parámetros `'C':[0.001, 0.01, 1, 10, 100, 1000] y 'gamma':[0.0001,0.001, 0.01, 0.1,1, 10,100]}.`

Para este caso el mejor juego de parámetros resulta `{'C': 100, 'gamma': 0.001, 'kernel': 'rbf'}` con accuracy de 0.80.

Mientras que pasándole al mismo modelo las Componentes Principales, los resultados para la mejor combinación de parámetros es `{'C': 1000, 'gamma': 0.001, 'kernel': 'rbf'}` también con un accuracy de 0.80, pero mejorando levemente la matriz (derecha):

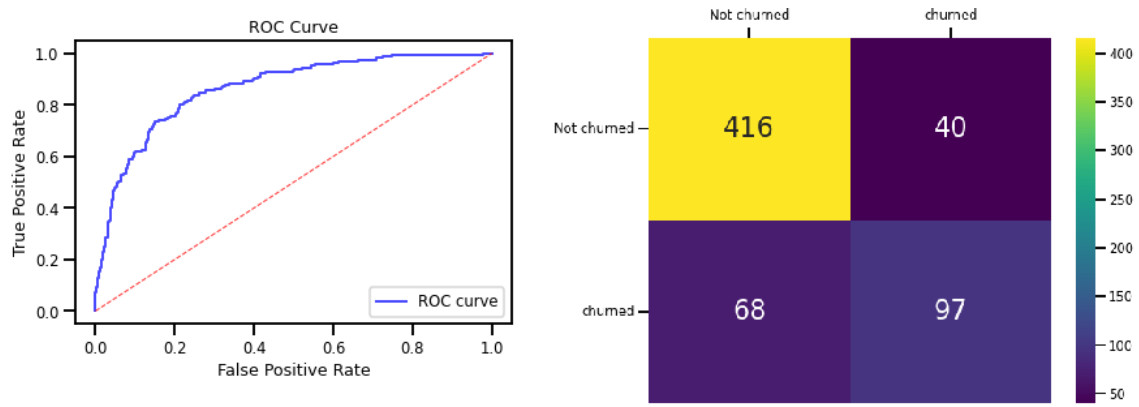


Regresión logística

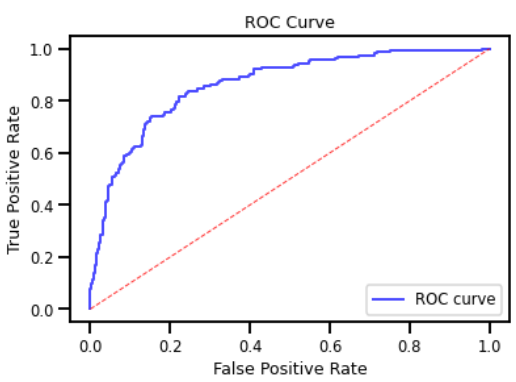
Considerando un dataset donde la respuesta por defecto cae dentro de 1 de 2 categorías (sí o no), en lugar de predecir esta respuesta directamente, una regresión logística modela las probabilidades de que una etiqueta Y pertenezca a una categoría particular. (An Introduction of Statistical Learning, pg. 133 - 2nd Ed. 2021, Tibshirani, Witten, Hastie, James).

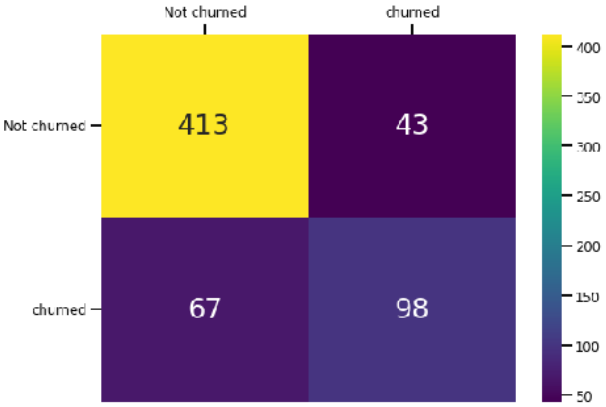
Mediante este método y tomando tanto los datos escalados como los datos con dimensionalidad reducida (y escalados), se inicia el modelo y se visualizan los resultados:

- Sin PCA: AUC de 0.864 y Accuracy de 0.826



- Con PCA: AUC de 0.863 y Accuracy de 0.823





Se puede notar prácticamente un mismo valor de área bajo la curva ROC y también leves diferencias en el valor obtenido de Accuracy y de resultados de la matriz de confusión.

Red neuronal

Estos modelos descomponen las entradas en capas de abstracción. Una red neuronal toma un vector de variables X=(X1, X2, … , Xp) como input y construye una función f(X) no lineal para predecir la respuesta de Y. (An Introduction of Statistical Learning, p. 404 - 2nd Ed. 2021, Tibshirani, Witten, Hastie, James). Su comportamiento está definido por la forma en que se conectan sus elementos individuales, así como por la ponderación de dichas conexiones. Estos pesos se ajustan automáticamente durante el entrenamiento de acuerdo con una regla de aprendizaje especificada hasta que la red neuronal lleva a cabo la tarea deseada.

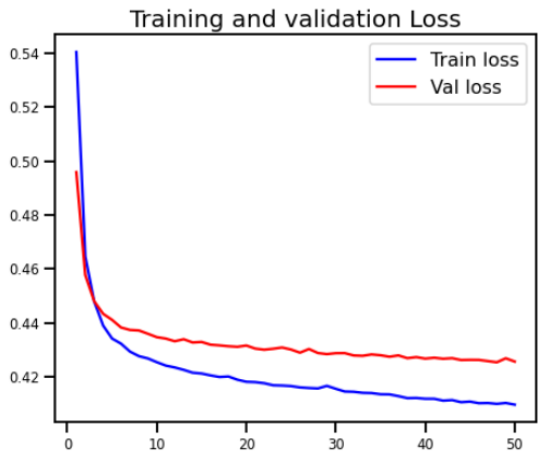
Para este método se utilizan las librerías y módulos de Keras - Tensorflow, ambas de código abierto, desarrolladas para la creación de modelos de aprendizaje automático y redes neuronales.

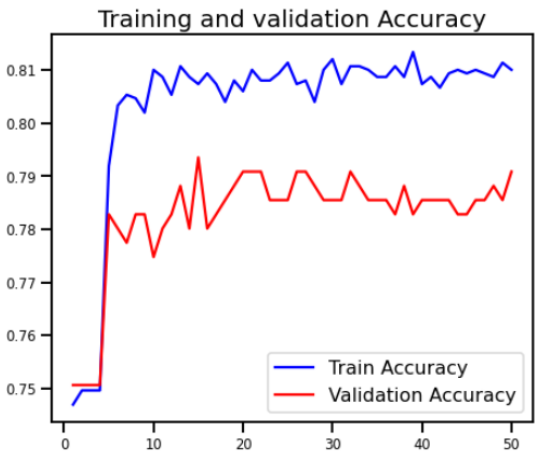
La arquitectura de la red fue definida luego de probar diseños más simples y más complejos, con o sin capas ocultas, y variando la cantidad de neuronas por capa, encontrándose mejor rendimiento para los casos con una primer capa de 10 neuronas con función de activación sigmoide, una hidden layer de 4 neuronas con función de activación ReLu, y en la capa final una neurona sigmoide para la selección binaria.

La función de pérdida establecida es la ‘binary_crossentropy’, con un optimizador de Adam y un Learning Rate de 0.001. También se definió un Batch Size de 5, una cantidad de 50 epochs de iteración para el entrenamiento, y un validation split de 0.2. Estos últimos hiperparámetros también han sido probado frente a otros, encontrando resultados más estables con los definidos (salvo la binary cross-entropy que es la indicada para este problema de clasificación).

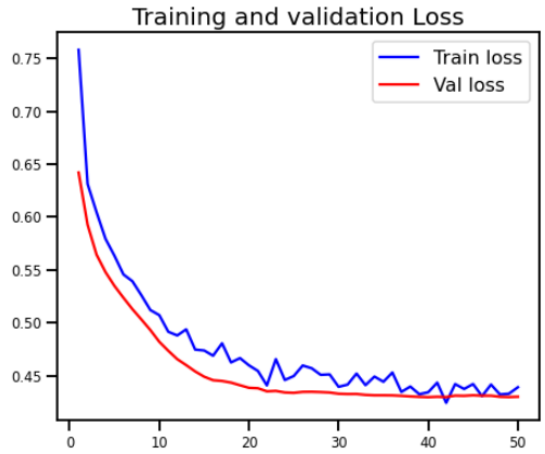
Al entrenar el modelo y observar la performance de los resultados predichos, se pudo notar un sobreajuste en el que el Accuracy de Train aumenta en mayor medida que el de Test, a la vez que la Train Loss baja más pronunciadamente que la Loss de Test, situación que se intenta aplacar a través de un drop-out del 50% de neuronas en la capa intermedia:

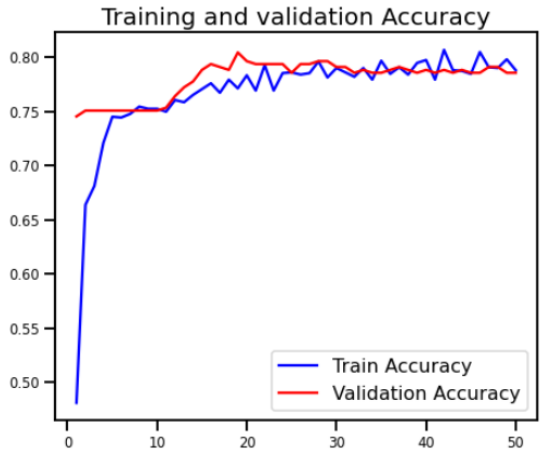
Sin drop-out



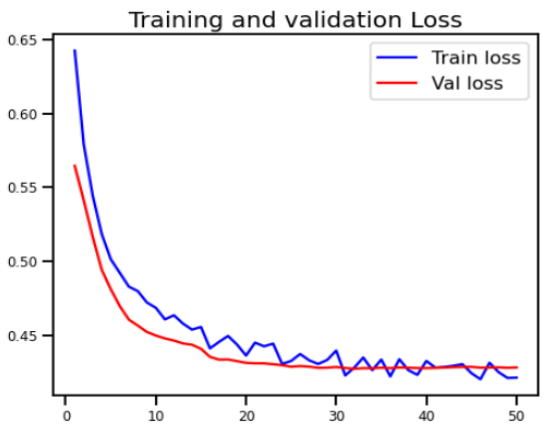


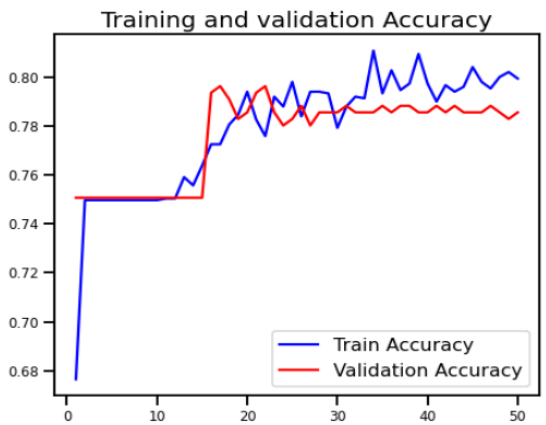
Agregando drop-out





De esta manera se han encontrado Test Accuracy de entre 0.79 y 0.83, con un Accuracy de Train también promediando 0.80 para los modelos entrenados con datos sin reducir dimensionalidad. La performance de la red para los datos provenientes del PCA disminuye, al hallar en este caso un sobre-ajuste algo más pronunciado y resultados más irregulares:





Conclusiones

Habiendo desarrollado cada uno de los modelos, se puede ver la comparativa de resultados dentro de la siguiente tabla de Accuracy:

Accuracy	SVC con kernel lineal	SVC kernel no lineal	Regresión Logística	Red Neuronal
Sin PCA	0.80	0.80	0.826	0.826
Con PCA	0.80	0.80	0.823	0.816

de la que se puede observar que el mejor rendimiento fue encontrado en los modelos de Regresión Logística y en la Red neuronal sin reducción de dimensionalidad, ya que por un margen reducido fueron las que obtuvieron mayor accuracy a la hora de predecir la variable Churn y serían los modelos seleccionados para este dataset.

Cabe destacar que el diseño de la red neuronal para los datos con y sin PCA es la misma, sin embargo a pesar de que las nuevas componentes (con solo 1 variable menos) guardan más de un 95% de varianza del dataset original, la red se empieza a comportar más erráticamente durante el entrenamiento y vuelve a sobre-ajustar; al igual que se han encontrado curvas de Loss y Accuracy algo distintas para mayor o menor cantidad de datos provenientes del pre-processing, por lo que se puede suponer que cambios en los datos de entrada aunque sean menores pueden llegar a afectar a la performance del modelo y que por esto puede resultar más conveniente optar por uno que generalice mejor como el caso de este Logistic Regression, o bien mejorar la arquitectura de la red para que devuelva mejores y más estables resultados.