

RNS: Churn

Martín Quijano - Martina Coletto

Información Básica del Dataset

- Tipos de datos: 13 columnas numéricas (12 enteras,) y 1 columna objetivo (Churn)
- Valores faltantes: No hay valores faltantes en el dataset

Variables

- | | |
|-----------------------|---------------------------|
| • Call Failure | • Distinct Called Numbers |
| • Complains | • Age Group |
| • Subscription Length | • Tariff Plan |
| • Charge Amount | • Age |
| • Seconds of Use | • Customer Value |
| • Frequency of use | • Churn |
| • Frequency of SMS | |

bajamos las variables status, age y age group.

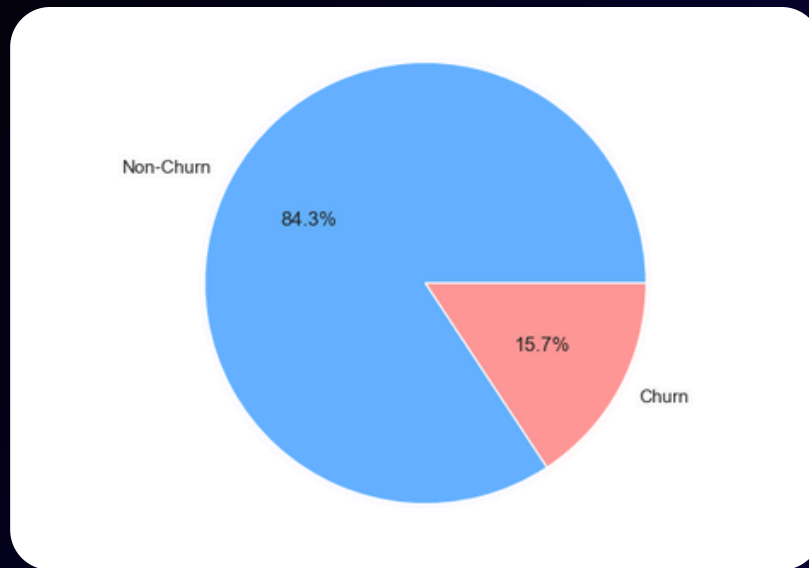
Status por la diferencia de 3 meses entre status y churn

Age group estaba como label encoding, y deberíamos haberla pasado a one hot encoding, pero hubiesen quedado demasiadas variables.

Despues de hacer el trabajo nos dimos cuenta que lo mismo deberíamos haber hecho con Charge Amount.

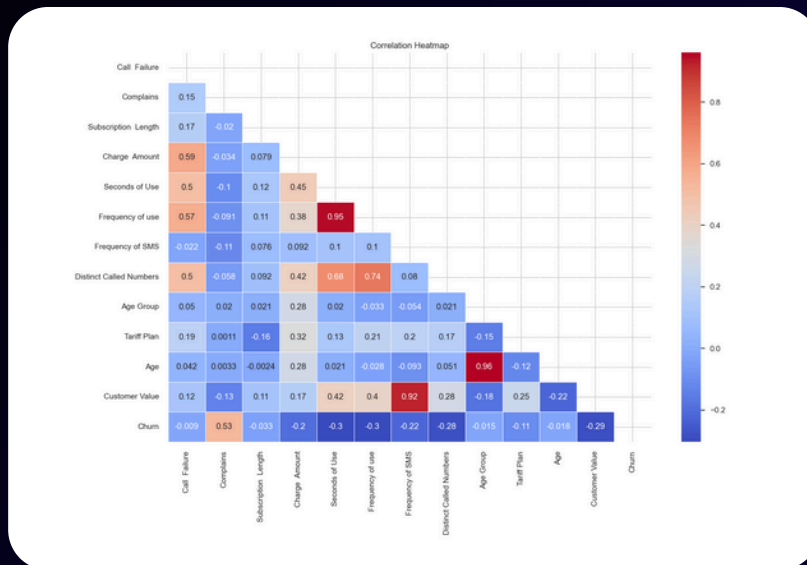


Análisis exploratorio

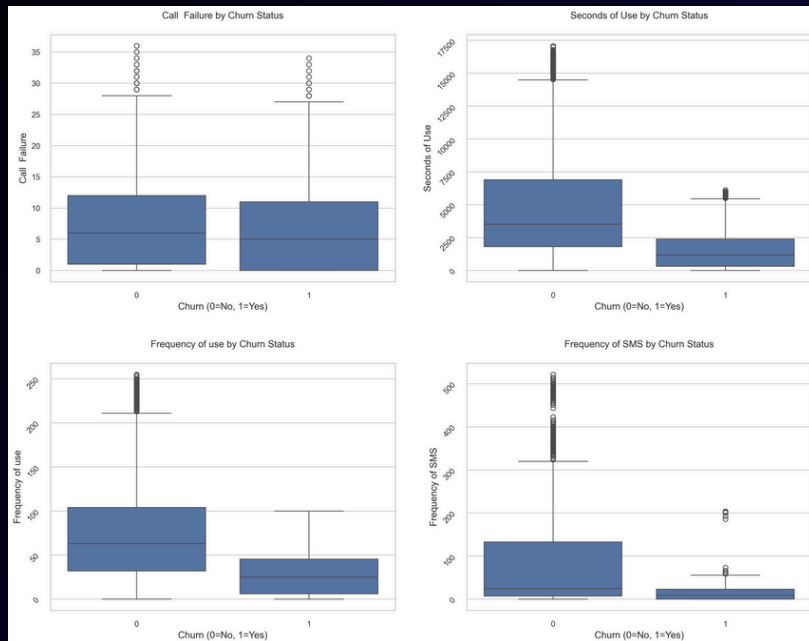


La muestra esta balanceada en sus clases.

Esto nos parecio muy importante, ya que si hacemos un modelo que solamente prediga no-churn, ese modelo tendria un accuracy de 84.3% de accuracy. Este numero es importante para que comparemos el accuracy del modelo, ya que a cualquier modelo que hagamos le vamos a exigir como minimo un accuracy de 84.3%

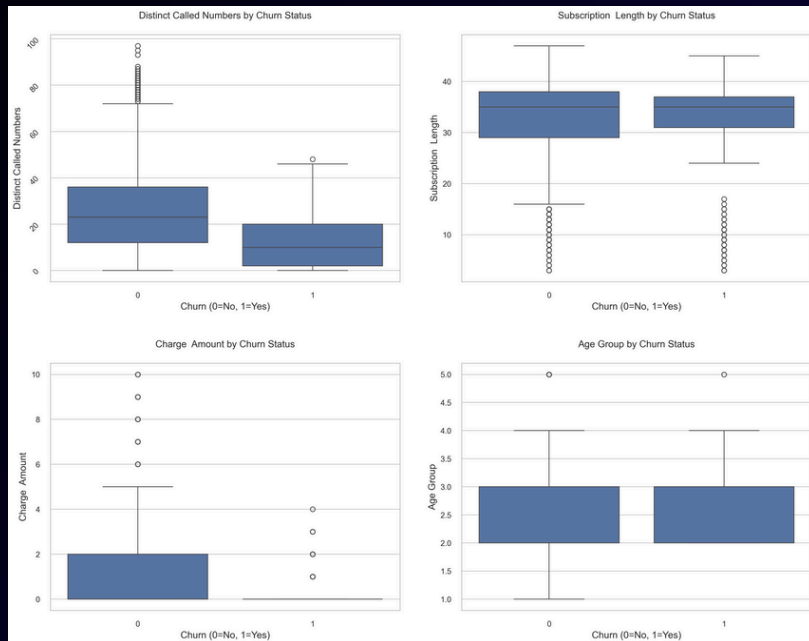


La muestra esta balanceada en sus clases Vemos que no hay correlaciones tan fuertes entre la variables. Las mayores son frequency of use con seconds of use, cosa que tiene mucho sentido. Age con age group tambien tiene sentido que esten correlacionadas, ya que hablan de lo mismo. La correlacion que nos sorprendio fue la de customer value con frequency of use.



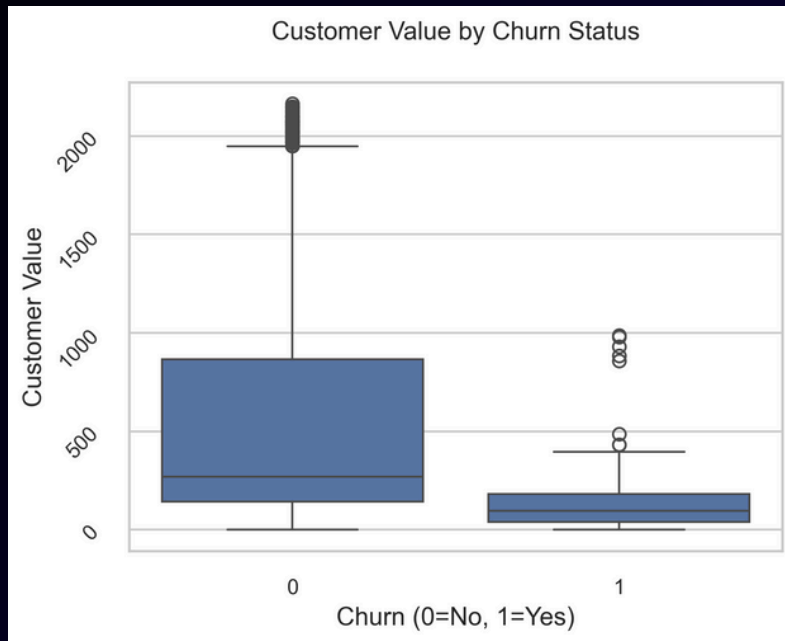
En call failure no vemos tanta diferencia.

En seconds of use, frequency of use y frequency of sms, vemos que mayores valores implican menos churn.

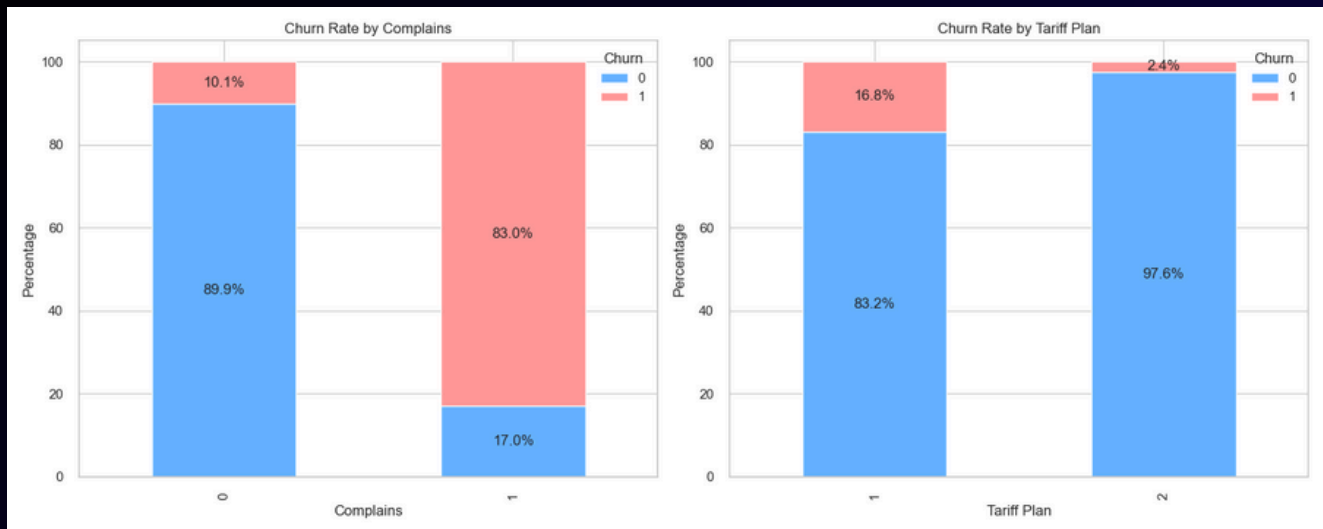


usamos boxplots para analizar la distribución de cada variable por churn. Esto nos permitió identificar diferencias significativas en la dispersión de las variables entre usuarios que hacen churn o no.

Vemos que los usuarios que hacen churn tienen menos llamadas realizadas y pagan menos, pero no hay diferencia entre usuarios que hacen churn o no en edad y largo de suscripción.



Vemos que los clientes mas valiosos hacen menos churn



con este grafico de barras podemos ver como afecta en las variables categorias los valores al churn. Vemos que los usuarios que realizan quejas son mas propensos a hacer churn, lo que habla de un mal servicio del call center. Por otro lado, vemos que en el tariff plan los usuarios que estan en el plan de "pay as you go" son mas propensos a churn, mientras que los usuarios de "contractual" hacen menos churn. Esto habla de usuarios mas "fidelizados" que deciden pasarse a un plan contractual en vez de pagar mientras consumen.



Arquitectura de la red neuronal

Arquitectura detallada

capa	neuronas	Activacion	Regularizador
entrada	-		-
capas ocultas	16	Relu	Linear (L1)
capa oculta	8	Relu	
capa de salida	1	Sigmoid	

Capas ocultas: tenemos 2 capa oculta, con 16 neuronas la primera y 8 la segunda. Utilizamos la función de activación relu.

También utilizamos el regularizador L1 para hacer feature importance.

Salida: nuestra capa de salida tiene 1 neuronas con activacion sigmoid. Solo necesitamos una neurona porque la salida es la probabilidad de churn

En las capas ocultas, al ser un dataset con tantas variables, decidimos usar tantas neuronas.

Parámetros de entrenamiento

- Función de pérdida: `binary_crossentropy`
- Optimizador: Adam
- Épocas: 300
- Tamaño del batch: 32
- Validación: split 0.2

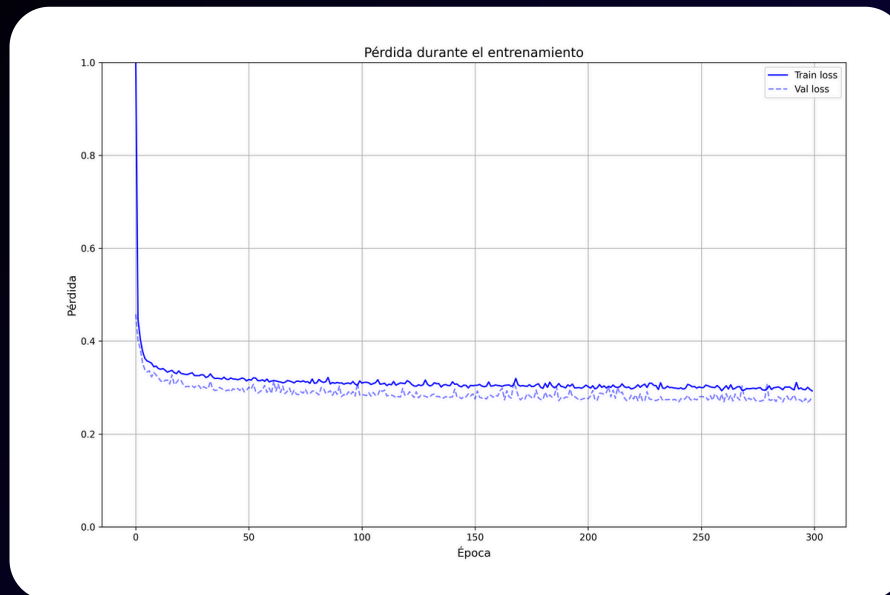
usamos binary crossentropy, porque la variable de respuesta es binaria (churn / no churn).

De batch size usamos 32, creemos que es un buen numero considerando que la muestra es de mas de 3500 datos.

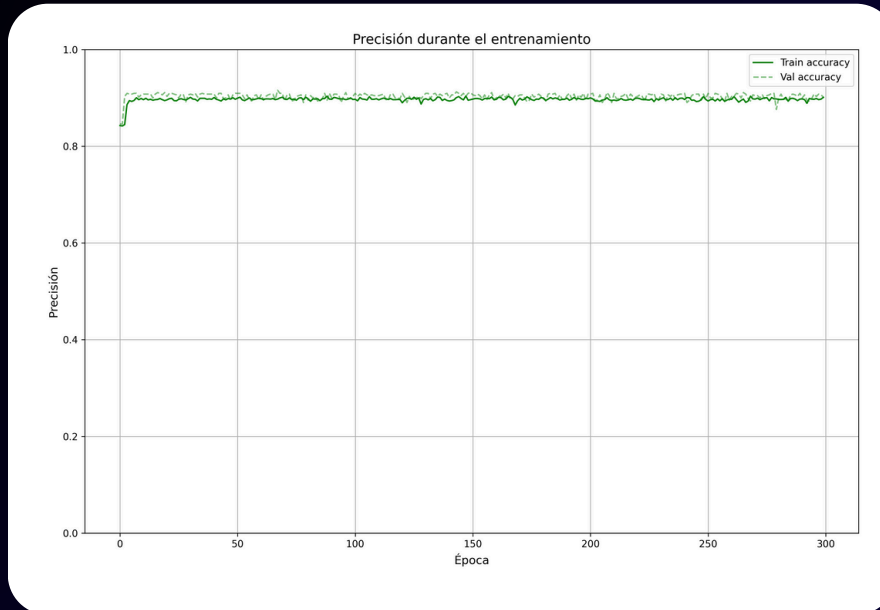


**Modelo
completo**

primero, arrancamos haciendo el modelo usando las 4 variables. Este modelo tiene regularizador L1



Aca vemos el loss del modelo. Vemos que converge a valores muy bajos, lo que indica que el modelo es bueno.

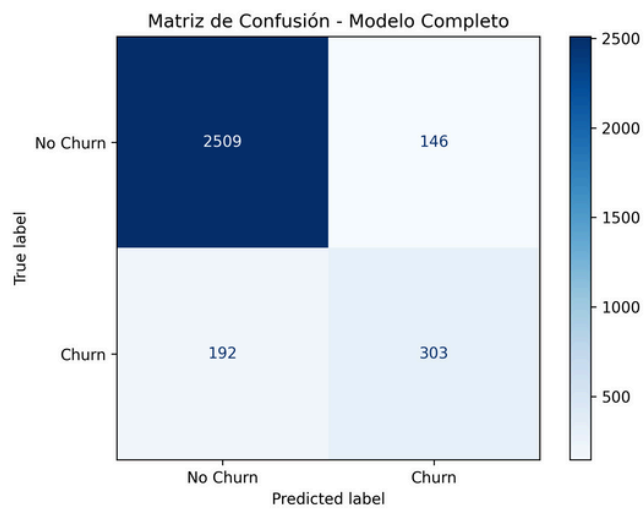


Aca vemos el accuracy del modelo. Vemos que converge a valores muy altos, lo que indica que el modelo es bueno.

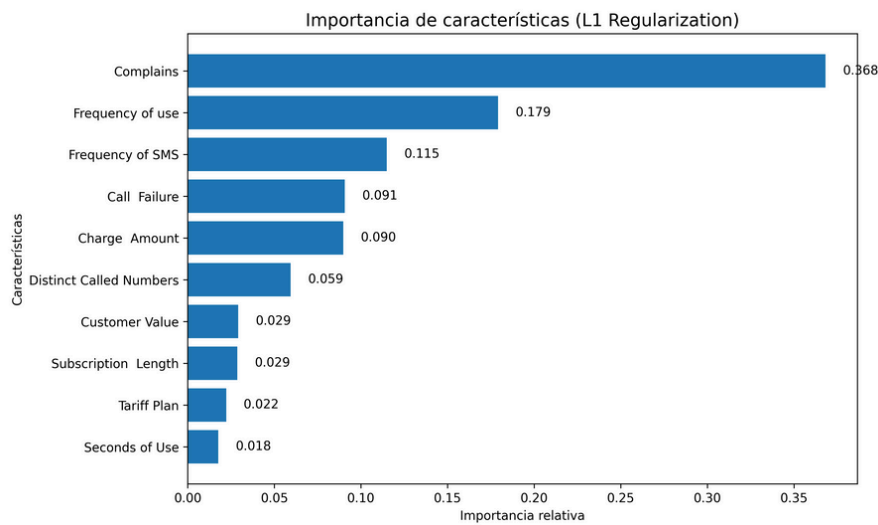
Métricas del Modelo

Accuracy	F1 Score	Precision	Recall
0.8927	0.6419	0.6748	0.6121

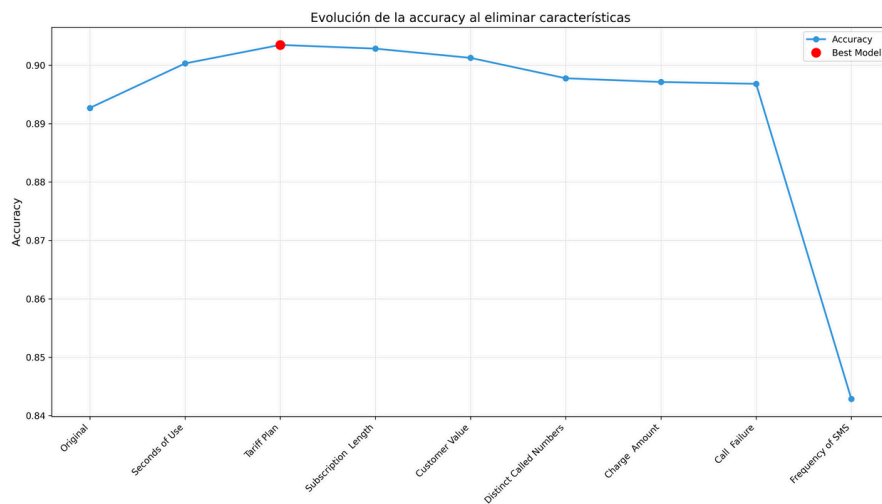
Estas son las metricas del modelo. Vemos que en accuracy esta muy bien, pero en el resto no. Esto se debe al desbalanceo del modelo



Por el desbalance de la muestra, la matriz de confusion muestra mas verdaderos positivos. No vemos un desbalanceo muy grande entre falsos positivos y falsos negativos.



Quisimos hacer un analisis de sensibilidad. Para eso, primero ordenamos las variables de mas a menos importante y hacemos el modelo. Una vez hecho el modelo, medimos y guardamos accuracy, f1, recall y precision. Luego, sacamos del modelo la variable menos importante. Con la nueva cantidad de variables, evaluamos el modelo, guardamos las metricas, volvemos a calcular la importancia de las variables, y eliminamos la menos importante. Repetimos ese ciclo hasta evaluar un modelo con solamente dos variables.



Este fue el resultado de ese proceso. Lo que encontramos es que se maximiza la accuracy cuando sacamos seconds of use y tariff plan. Igualmente, el accuracy no se ve mejorado significativamente (pasa de 0.89 a 0.9).

Esto nos indica también que podemos sacrificar accuracy por simplicidad usando el modelo de 3 variables.

Métricas del Modelo

Accuracy	F1 Score	Precision	Recall
0.9035	0.6122	0.8304	0.4848

Estas fueron las métricas del mejor modelo que vimos en la slide anterior. Vemos que el accuracy mejora levemente, pero en precision hay una mejora significativa.

Métricas del Modelo

Accuracy	F1 Score	Precision	Recall
0.8429	0.0000	0.0000	0.0000

Estas fueron las metricas del modelo con dos variables que vimos en la slide anterior.

Vemos que empeoran todas las metricas. Nos hubiese gustado ver las metricas del modelo de 3 variables.



Conclusiones

Como conclusion, podemos decir dos cosas.

La primera es que estos modelos tan complejos no mejoran significativamente el modelo que planteamos al principio.

Un modelo que predice no churn todo el tiempo implica un accuracy de 84.3%, mientras que nuestro mejor modelo tiene un accuracy de 90%. Es decir, implementar este modelo tan complejo solo trae una mejora del 6% de accuracy.

Como recomendacion a la empresa, lo que podemos decir es que por lo visto en los boxplots, el servicio que ofrecen es bueno (a mas uso en el call center, menor churn). Se podrian destinar recursos a la difusion de la existencia del call center para los clientes.

valores reales	no-churn	2268	387
	churn	334	161
		no-churn	churn
		valores predichos	
SUMA ERR CUADRADOS	486.1494		
CANTIDAD DE ERRORES	3150		
MSE	0.154333		
MEAN BCE	0.335207		
accuracy	77.11%		

Antes

Solver

valores reales	no-churn	2655	0
	churn	495	0
		no-churn	churn
		valores predichos	
SUMA ERR CUADRADOS	494.7148		
CANTIDAD DE ERRORES	3150		
MSE	0.157052		
MEAN BCE	0.000327		
accuracy	84.29%		

Despues

con solver, redujimos el mean binary cross-entropy (mean bce). El primer modelo tiene weights random, y el segundo esta con solver.

El solver determino que lo mejor era predecir siempre no churn para reducir el BCE.

Gracias

Martín Quijano - Martina Coletto