Estudio comparativo de clasificadores para la predicción de fuga de clientes en la empresa Orange Telecom

Bastián Barraza[1] y Ricardo Menares[2]
[1]Estudiante en Ingeniería en Estadística y Ciencia de Datos, Universidad de Valparaíso, Valparaíso, Chile.
[2]Estudiante en Ingeniería en Estadística y Ciencia de Datos, Universidad de Valparaíso, Valparaíso, Chile.

RESUMEN

Uno de los principales problemas que se enfrentan las empresas es al de la fuga de clientes, debido a que influye en la pérdida directa de los ingresos de una institución. Además, al momento de reemplazar un cliente que se desvinculó de una empresa, hay que invertir en marketing para poder atraer nuevos usuarios, por lo que es notorio que es más costoso y difícil obtener nuevos clientes que, por el contrario, retener a los clientes que podrían irse.

En este contexto, el objetivo principal de esta investigación es realizar una comparación entre diferentes clasificadores para identificar y predecir clientes que abandonan la empresa Orange Telecom, institución de telecomunicaciones, según varias variables importantes que estos usuarios poseen en la empresa, tales como la cantidad de llamadas que realiza, minutos usados, entre otras.

Según los resultados obtenidos a través de distintas métricas de desempeño de la matriz de confusión de cada clasificador, se puede deducir que el modelo Gradient Boosting fue el que obtuvo mejor desempeño para poder clasificar la fuga de clientes de la empresa, seguido por clasificadores como Random Forest y XGBoost, tanto antes y después de realizar el análisis de componentes principales.

PALABRAS CLAVES

Clasificadores, fuga de clientes, matriz de confusión, análisis de componentes principales.

I. INTRODUCCIÓN

Los clientes son muy importantes para toda empresa, pues son el principal sustento financiero que posee. En este contexto, uno de los objetivos que debe tener una compañía es el de poder conocer a sus clientes y saber evaluar si están a gusto con los servicios de la empresa, o más específicamente, si estos usuarios pueden abandonar la empresa en el futuro. Lo anterior es sumamente importante, debido a que, si un usuario abandona la compañía, pierde los recursos financieros que obtenía de ese cliente, más aún, Miranda, J & et al. (2005) indican que los clientes son uno de los activos más importantes para una institución financiera, ya que está estrechamente relacionada con las utilidades del negocio.

A causa de esto, la empresa Orange Telecom (empresa de telecomunicaciones de Estados Unidos) le motiva conocer un método eficiente para predecir si un usuario potencialmente pueda abandonar la empresa, a modo de poder retenerlo antes que efectúe el retiro, ya que resulta ser más costoso conseguir nuevos clientes que retener a los que ya están en la empresa.

Cabe destacar, que según la empresa que se esté evaluando variarán los servicios que le interesan al cliente, según Molina (2009) estos servicios en empresas de telecomunicaciones comúnmente son las de tráfico de larga distancia, telefonía local, internet, cargos de acceso, servicios privados, factoras, cuentas corrientes, contratos, entre otros.

En consiguiente, para poder resolver el problema descrito, se usan los datos proporcionados por la empresa Orange Telecom de sus clientes y en base a las características de estos, se implementan 12 diferentes clasificadores bastante conocidos por la comunidad científica, tales como el LDA, QDA, Árbol de decisión, Random Forest, Regresión Logística, SVC lineal, SVC radio basal, Perceptrón, Perceptrón multicapa, K-Neighbors Classifier, Gradient Boosting y el XGBoost. Dado que se ocupan bastantes clasificadores, el objetivo principal del estudio es el de poder comparar y luego elegir uno o más clasificadores que mejor se adaptaron a los datos de la compañía para poder clasificar si un cliente abandonará la empresa o no, de modo que ayude a Orange Telecom a retener a estos usuarios.

Para realizar lo anterior, se aplica la metodología KDD que según Timarán-Pereira, S. & et al. (2016) es básicamente un proceso automático en el que se combinan descubrimiento y análisis. El proceso consiste en extraer patrones en forma de reglas a partir de los datos para que el usuario los analice. Además, señala que generalmente esta tarea tiene 5 etapas, las cuales son: selección de datos, preprocesamiento y limpieza de datos, transformaciones y reducciones, minería de datos y por último interpretar y/o evaluar los resultados.

II. MATERIALES Y MÉTODOS

A. MATERIALES

El conjunto de datos obtenido de Orange Telecom consiste en datos de la actividad de los clientes, junto con una etiqueta de abandono que indica si el cliente canceló la suscripción al servicio. Están disponibles 2 bases de datos, catalogados como "churn-bigml-20" con 20 columnas y 667 filas y "churn-bigml-80" con 20 columnas y 2666 filas. En total se tienen 20 variables y la suma de 3.333 datos.

Las variables que contiene estas bases de datos son las siguientes: State, Account length, Area code, International plan, Voice mail plan, Number vmail messages, Total day minutes, Total day calls, Total day charge, Total eve minutes, Total eve calls, Total eve charge, Total night minutes, Total night calls, Total night charge, Total intl minutes, Total intl calls, Total intl charge, Customer service calls y la variable de respuesta Churn.

B. METODOS

Según la metodología elegida para trabajar con esta base de datos (KDD) y teniendo en posesión los datos otorgados por Orange Telecom, se realiza un preprocesamiento y limpieza de datos. Se pudo observar que la base de datos no tenía problemas de datos faltantes ni datos mal escritos. Por lo tanto, se considera la base de datos limpia y consistente.

Por otro lado, en la etapa de transformación, es necesario recalcar que esta base de datos no fue creada para poder trabajar con modelos de clasificación, por tanto, si fue necesario realizar una transformación a los datos. En primer lugar, se debió cambiar los valores de las variables categóricas binarias que contenían en sus observaciones los datos de “Yes” y “No”, tales como la variable de si el usuario tiene plan internacional o no, también si tiene plan de correo de voz y también la variable de respuesta, que indica si el cliente abandona la empresa o no, por números entre 1 y 0, respectivamente. Lo anterior, se desarrolló debido a que varios clasificadores no sirven para trabajar con variables categóricas, entorpeciendo los resultados que se consiguen.

En consiguiente, se detectaron valores atípicos en varias variables, tales como en los totales de minutos usados, recargas y llamadas tanto como para las secciones diarias, noches, internacionales y festivos. Además, es preciso destacar que entre las distintas variables hay una gran variabilidad en sus datos, pues hay variables binarias hasta variables continuas y en diferentes medidas, por lo tanto, la importancia de los atributos al momento de entrenar al clasificador es distinta según la variable. Para solucionar todo lo anterior, se hizo un trabajo de reescalamiento de los datos, en donde se centraron los datos de cada variable en 0 y su desviación estándar en 1 con:

$$Z = \frac{X - \bar{X}}{S_x}$$

Donde X corresponde al dato, $\bar{X}$ al promedio de los datos según variable y S_x a la desviación estándar.
De esta manera, se logra cuantificar las distintas variables de tal modo de minimizar los problemas de la importancia de los distintos datos.

En consiguiente, es preciso destacar que se realizó el ACP (Análisis de Componentes Principales) para poder reducir la dimensionalidad de las variables sin perder tanta variabilidad en la respuesta de la variable de fuga de cliente. Posteriormente, se compara los distintos clasificadores con y sin ACP para poder visualizar su comportamiento cuando se reducen las variables que aportan menor variabilidad en la variable de respuesta.

Finalizando, en esta investigación se tomaron en cuenta 12 distintas métricas de desempeño, por lo tanto, se explica en términos simples en que consiste cada una:

1. LDA: El Linear Discriminant Analysis es un método de clasificación supervisado en el que dos o más grupos son conocidos a priori y nuevas observaciones se clasifican en uno de ellos en función de sus características. Ocupando el teorema de Bayes y estimando los parámetros de la distribución a priori con el estimador máximo verosímil, se estima la probabilidad de que una observación dado un valor de los predictores pertenezca a alguna clase. Además, este clasificador trabaja bajo el supuesto de que las k clases siguen una distribución normal y con una matriz de varianza-

covarianza iguales para todas las clases. La función de discriminante en este modelo es:

$g_k(x)=x^T\sum^{-1}\mu_k-\frac{1}{2}\mu_k^T\sum^{-1}\mu_k+ln(\lambda_k)$

Notar que esta función de discriminante es lineal con respecto a X, por eso se llama "Linear" Discriminant Analysis. La regla de decisión es:

d(x)= argmáx(g_k(x)), k=1,2,...,K.

2. QDA: El Quadratic Discriminant Analysis, al igual que el LDA, ocupa el método bayesiano antes descrito para poder clasificar. Sin embargo, este clasificador considera que cada clase k tiene su propia matriz de covarianza. A raíz de lo anterior, se obtiene una función de discriminante cuadrática en vez de lineal, siendo:

$g_k(x)=-\frac{1}{2}\ln(\sum_k)-\frac{1}{2}(x-\mu_k)^T\sum_k^{-1}(x-\mu_k)+ln(\mu_k)$

Y la regla de clasificación está dada por:

d(x)=argmáx(g_k(x)) ; k=1,2,...,K.

Cabe destacar que QDA genera límites de decisión curvos por lo que puede aplicarse a situaciones en las que la separación entre grupos no es lineal.

3. Árbol de decisión: Los árboles de decisión son representaciones gráficas de posibles soluciones a una decisión basadas en ciertas condiciones, además de ser un algoritmo de aprendizaje supervisado y pueden realizar tareas de clasificación o regresión. La comprensión de su funcionamiento suele ser simple y a la vez muy potente.

Además, los árboles de decisión tienen un primer nodo llamado raíz (root) y luego se descomponen el resto de atributos de entrada en dos o más ramas planteando una condición que puede ser cierta o falsa. Se bifurca cada nodo en las ramas y vuelven a subdividirse hasta llegar a las hojas que son los nodos finales y que equivalen a respuestas a la solución: Si/No, Comprar/Vender, o lo que sea que se esté clasificando.

4. Random Forest: Este método es una combinación de árboles predictores tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos, además, trabaja con una combinación de árboles de decisión combinados con bagging. Al usar bagging, distintos árboles ven distintas porciones de los datos. Ningún árbol ve todos los datos de entrenamiento. Esto hace que cada árbol se entrene con distintas muestras de datos para un mismo problema. De esta forma, al combinar sus resultados, unos errores se compensan con otros y se obtiene una predicción que generaliza mejor.

Más específicamente, toma Una muestra de los N casos del conjunto de entrenamiento aleatoriamente, pero CON reemplazo. Esta muestra será el conjunto de entrenamiento para construir el árbol i-ésimo. Luego, si existen M variables de entrada, un número m<M se especifica tal que para cada nodo, m variables se seleccionan aleatoriamente de M. La mejor división de estos m atributos es usado para ramificar el árbol. El valor m se mantiene constante durante la generación de todo el bosque. Luego cada árbol crece hasta su máxima extensión posible y genera una clasificación.

Así, cada árbol genera una clasificación y el resultado final será la clase con mayor número de clasificaciones de todo el bosque.

5. Regresión Logística: la regresión logística es un tipo de regresión utilizado para predecir el resultado de una variable categórica (principalmente en variables binarias) en función de las variables predictoras. El análisis de regresión logística se enmarca en el conjunto de Modelos Lineales Generalizados. Las probabilidades que describen el posible resultado de un único ensayo se modelan como una función de variables explicativas, utilizando una función logística.

Debido a que al generar una recta (hiperplano si hay múltiples variables), se pueden obtener valores predichos distintos de 0 y 1, este modelo entraría en contradicción con la definición de la variable respuesta binaria. Por tanto y para evitar estos problemas, la regresión logística transforma el valor devuelto por la regresión lineal con una función cuyo resultado siempre está comprendido entre 0 y 1. Una de esas funciones es la siguiente (conocida como sigmoide):

Sigmoide= $\frac{1}{1+e^{-y}}$

Luego, Una vez obtenido las estimaciones de los coeficientes del modelo (β_0, β_1,..., β_n) se puede obtener la probabilidad de que una nueva observación pertenezca a la clase y=1 con la ecuación:

$P(Y=1|X=x)=\frac{e^{(b_0+b_1X_1+..+b_nX_n)}}{1+e^{(b_0+b_1X_1+..+b_nX_n)}}$

6. SVC Lineal: el SVC (Support Vector Classifier) lineal es un clasificador de aprendizaje supervisado, en donde construye un hiperplano o conjunto de hiperplanos en un espacio de dimensionalidad alto (incluso infinito). Una buena separación entre las clases permitirá una clasificación correcta.

Específicamente, este tipo de algoritmos buscan el hiperplano que tenga la máxima distancia (margen) con los puntos de las diferentes clases. De esta forma, los puntos del vector que son etiquetados con una categoría estarán a un lado del hiperplano y los casos que se encuentren en la otra categoría estarán al otro lado. Así, los vectores de soporte son los puntos que definen el margen máximo de separación del hiperplano que separa las clases.

Además, este método ocupa un kernel (K) que es una función que devuelve el resultado del producto escalar entre dos vectores realizado en un nuevo espacio dimensional distinto al espacio original en el que se encuentran los vectores. Si se obtiene el kernel, también se obtienen directamente los vectores soporte (y el hiperplano) en la dimensión correspondiente al kernel. Ha esto se le suele conocer como kernel trick porque, con solo una ligera modificación del problema original, se puede obtener el resultado para cualquier dimensión. Existen multitud de kernels distintos, para el modelo SVC lineal es:

$K(x,x^T)=xx^T$

7. SVC Radio Basal: Este algoritmo es un caso especial de los Support Vector Classifier, en donde este método consume más tiempo, pero generalmente otorga mejor predicción que el SVC lineal y además es más flexible.

La diferencia matemática entre este método y el lineal es en gran medida a su kernel, siendo el siguiente para el modelo SVC radio basal:

$K(x,x^T)=\exp(-\lambda||x-x^T||^2)$

Notar que el valor de $\lambda\lambda$ controla el comportamiento del kernel, cuando es muy pequeño, el modelo final es equivalente al obtenido con un kernel lineal y a medida que aumenta su valor, también lo hace la flexibilidad del modelo.

8. Perceptrón: El Perceptrón es un algoritmo de aprendizaje para tareas de clasificación binaria por lo que nos permite clasificar al conjunto de entradas únicamente dentro de 2 posibilidades y además solo ocupa una capa. En consiguiente, puede clasificarse como uno de los más simples modelos de las redes neuronales artificiales.

Consiste en un único nodo o neurona que toma una fila de datos como entrada y predice una etiqueta de clase. Esto se consigue calculando la suma ponderada de las entradas y un sesgo (establecido en 1). La suma ponderada de la entrada del modelo se denomina activación.

La neurona recibe un conjunto de entradas ($x=x_1, x_2, .., x_n$) y a cada una se le asocia un peso ($w=w_1, w_2, .., w_n$) que les dará un nivel de importancia a cada entrada. Luego se define z como la sumatoria ponderada de cada entrada por su correspondiente peso:

$Z=\sum_{i=1}^{n} b + w_i x_i$

Luego, para determinar una salida, se debe definir a z como una función de activación. Por ejemplo, si está por encima de 0, el modelo dará como resultado un 1, de lo contrario, dará 0. Cabe destacar, que un perceptrón solamente es capaz de trabajar para aquellos casos donde las muestras son line

9. Perceptrón Multicapa: El perceptrón multicapa es una red neuronal artificial (RNA) formada por múltiples capas, de tal manera que tiene capacidad para resolver problemas que no son linealmente separables, lo cual es la principal limitación del perceptrón antes mencionado.

La arquitectura del Perceptrón multicapa se caracteriza porque tiene sus neuronas agrupadas en capas de diferentes niveles. Cada una de las capas está formada por un conjunto de neuronas y se distinguen tres tipos de capas diferentes: la capa de entrada, las capas ocultas y la capa de salida.

Las neuronas de la capa de entrada no actúan como neuronas propiamente dichas, sino que se encargan únicamente de recibir las señales o patrones del exterior y propagar dichas señales a todas las neuronas de la siguiente capa. La última capa actúa como salida de la red, proporcionando al exterior la respuesta de la red para cada uno de los patrones de entrada. Las neuronas de las capas ocultas realizan un procesamiento no lineal de los patrones recibidos. Además, generalmente todas las neuronas de una capa están conectadas a todas las neuronas de la siguiente capa. Se dice entonces que existe conectividad total o que la red está totalmente conectada.

10. K-Neighbors Classifier: K-Neighbors Classifier es un algoritmo no paramétrico de clasificación de aprendizaje supervisado de Machine Learning. Sirve esencialmente para clasificar valores buscando los puntos de datos “más similares” (por cercanía) aprendidos en la etapa de entrenamiento y haciendo conjeturas de nuevos puntos basado en esa clasificación.

El espacio es particionado en regiones por localizaciones y etiquetas de los ejemplos de entrenamiento. Un punto en el espacio es asignado a alguna clase si esta es la clase más frecuente entre los k ejemplos de entrenamiento más cercano. Generalmente se usa la distancia euclidiana para identificar lo anterior:

$d(x_i,x_k)=\sqrt{\sum_{w=1}^{p}(x_{wi}-x_{wj})^2}$

Por tanto, se calcula la distancia entre los vectores almacenados y el nuevo vector, y se seleccionan los k ejemplos más cercanos. El nuevo ejemplo es clasificado con la clase que más se repite en los vectores seleccionados.

Este método supone que los vecinos más cercanos nos dan la mejor clasificación y esto se hace utilizando todos los atributos; el problema de dicha suposición es que es posible que se tengan muchos atributos irrelevantes que dominen sobre la clasificación: dos atributos relevantes perderían peso entre otros veinte irrelevantes. Para arreglar lo anterior

es posible darle un "peso" a cada atributo para darle más valor a los más importantes.

11. Gradient Boosting: Un modelo Gradient Boosting es un método no paramétrico que está formado por un conjunto de árboles de decisión individuales, entrenados de forma secuencial, de forma que cada nuevo árbol trata de mejorar los errores de los árboles anteriores. La predicción de una nueva observación se obtiene agregando las predicciones de todos los árboles individuales que forman el modelo.

Durante el entrenamiento, los parámetros de cada modelo de árbol son ajustados iterativamente tratando de encontrar el mínimo de una función objetivo, que puede ser la proporción de error en la clasificación, el área bajo la curva (AUC), la raíz del error cuadrático medio (RMSE) o alguna otra.

Este método ocupa el aumento de gradiente, que es un tipo de algoritmo de aumento. Se basa en la intuición de que el mejor modelo siguiente posible, cuando se combina con modelos anteriores, minimiza el error de predicción general. La idea clave es establecer los resultados objetivo para este próximo modelo con el fin de minimizar el error. El aumento de gradiente se puede utilizar tanto para clasificación como para regresión.

12. XGBoost: El modelo XGBoost (eXtreme Gradient Boosting) es una implementación de árboles de decisión con Gradient boosting diseñada para minimizar la velocidad de ejecución y maximizar el rendimiento. Este modelo, hoy en día es uno de los más usados en la comunidad científica para clasificar, debido a su alto desempeño y velocidad de trabajo.

Este modelo sigue el mismo principio del aumento de gradiente que el Gradient Boosting. Sin embargo, existen diferencias en los detalles de modelado. Específicamente, XGBoost utiliza una formalización de modelo más regularizada para controlar el sobreajuste, lo que le brinda un mejor rendimiento, por lo tanto, se puede deducir que es una versión mejorada del Gradient Boosting.

III. RESULTADOS Y DISCUSIÓN

A. Métricas de Desempeño

Para poder evaluar a los distintos clasificadores, se hace uso de la matriz de confusión. Esta matriz, es una herramienta que permite visualizar el desempeño de un clasificador empleado. La dimensión de esta matriz varía en la cantidad de clases que tiene la variable dependiente, si esta cantidad se denota por n, el tamaño de la matriz de confusión resultante será nxn. En el caso de la variable de fuga de cliente que se utiliza en este estudio, tiene dimensión n= 2 y por tanto el tamaño de la matriz de confusión es 2x2.

En consiguiente, la matriz de confusión al ser de dimensión 2x2 tendrá la siguiente estructura:

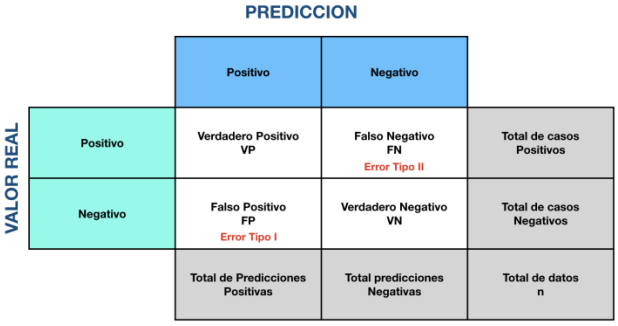

Figura 1: Diseño de matriz de confusión 2x2.

Donde VP y VN indican la cantidad de veces que el clasificador pudo identificar un resultado realmente positivo y negativo, respectivamente. FN y FP indican la cantidad de veces que el clasificador se equivocó, tanto para los casos negativos y positivos, respectivamente.

Luego, se pueden obtener distintas métricas de desempeño de esta matriz, que servirán para poder evaluar a los clasificadores. Las que se ocupan en este estudio son:

1. Accuracy: Indica lo cerca que está el resultado de una medición del valor verdadero. Así, se puede representar como la proporción de los resultados verdaderos (VP+VN en la matriz presentada) dividido por el número total de datos. Entonces, su cálculo es:

$$\text{Accuracy}=\frac{VP+VN}{n^{\circ}\ total\ de\ datos}$$

2. Recall: Es la proporción de casos positivos que fueron correctamente identificados por el clasificador. Por ende, se representa como la división entre los casos verdaderos positivos (VP) entre la suma de los casos verdaderos positivos + los casos falsos negativos (FN). Entonces, su cálculo se desarrolla como:

$$\text{Recall}=\frac{VP}{VP+FN}$$

3. Precisión: Es el cociente entre los casos positivos bien clasificados por el modelo y el total de predicciones positivas. En otras palabras, indica cuántos de los casos predichos correctamente resultaron ser realmente positivos. Se calcula de la siguiente manera:

$$\text{Precisión}=\frac{VP}{VP+FP}$$

4. Especificidad: Conocida también como “la tasa de verdaderos negativos”, indica la cantidad de casos negativos verdaderos que el algoritmo pudo clasificar de buena manera. Por lo tanto, se obtiene dividiendo los casos verdaderos negativos (VN) entre la suma de los casos verdaderos negativos (VN) y los falsos positivos (FP) de la siguiente manera:

$$\text{Especificidad}=\frac{VN}{VN+FP}$$

5. Medida F: Es la media armónica entre precisión y recall. Se calcula como:

$$\text{Medida F}=\frac{(2)(precisión)(recall)}{precisión+recall}$$

Con estas 5 métricas se evalúan los clasificadores en este estudio, lo que es suficiente para poder elegir cuales obtuvieron mejores resultados. Cabe destacar, que se calcularán estas métricas tanto a los clasificadores antes y después de realizar el ACP, para poder comparar los resultados.

B. Estudio descriptivo de los datos

Se muestra a continuación estadísticas descriptivas tradicionales tales como la media, mediana desviación estándar, valor mínimo y máximo y algunos cuartiles de las variables:

TABLA 1
TABLAS DE ESTADÍSTICAS DESCRIPTIVAS BÁSICAS

	Account length	Area code	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	Total eve calls	Total eve charge	Total night minutes	Total night calls
count	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000
mean	101.064806	437.182418	8.099010	179.775098	100.435644	30.562307	200.980348	100.114311	17.083540	200.872037	100.107711
std	39.822106	42.371290	13.688365	54.467389	20.069084	9.259435	50.713844	19.922625	4.310668	50.573847	19.568609
min	1.000000	408.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	23.200000	33.000000
25%	74.000000	408.000000	0.000000	143.700000	87.000000	24.430000	166.600000	87.000000	14.160000	167.000000	87.000000
50%	101.000000	415.000000	0.000000	179.400000	101.000000	30.500000	201.400000	100.000000	17.120000	201.200000	100.000000
75%	127.000000	510.000000	20.000000	216.400000	114.000000	36.790000	235.300000	114.000000	20.000000	235.300000	113.000000
max	243.000000	510.000000	51.000000	350.800000	165.000000	59.640000	363.700000	170.000000	30.910000	395.000000	175.000000

Total night charge	Total intl minutes	Total intl calls	Total intl charge	Customer service calls
3333.000000	3333.000000	3333.000000	3333.000000	3333.000000
9.039325	10.237294	4.479448	2.764581	1.562856
2.275873	2.791840	2.461214	0.753773	1.315491
1.040000	0.000000	0.000000	0.000000	0.000000
7.520000	8.500000	3.000000	2.300000	1.000000
9.050000	10.300000	4.000000	2.780000	1.000000
10.590000	12.100000	6.000000	3.270000	2.000000
17.770000	20.000000	20.000000	5.400000	9.000000

Se aprecia que hay bastante heterogeneidad en las estadísticas básicas de cada variable, pues cada una en general representa un rango de datos distinto. así, se puede observar que la media para, por ejemplo, el código de área es muy diferente a la media del número de mensajes por vmail. Lo anterior es lógico debido a que el código de área solo toma valores de 408, 415 y 510. Mientras que espectro de valores de la variable número de mensajes por vmail son bastante más pequeños.

Por otro lado, se aprecia que las variables del total de minutos y llamadas (tanto para día, eve y noche) tienen una media bastante significativa, en general mayor a 100 llamadas por el periodo respectivo. Por lo que se puede concluir que las personas de esta compañía ocupan bastante el servicio, además se destaca la desviación estándar de estas variables, pues igual son significativas. También, en la mayoría de las variables tienen como mínimo el valor 0, entonces hay personas en esta compañía que no estarían ocupando el servicio, además de que el máximo en varias variables es mayor a 350, lo que indica que hay personas que ocupan al menos 5 horas al día los minutos de la compañía.

En cuanto al plan internacional, se aprecia que este se ocupa mucho menos, pues la media del total de minutos es de solo de 10.2, con una desviación estándar de 2.7. Además, el número de llamadas en promedio se reduce a solo 4. Por lo demás se aprecia que el cuartil 1 y 2 de las variables que contemplan el plan internacional indican que apróx un 25% de las personas llaman entre 3 y 4 veces internacionalmente, y ocupan entre 8.5 y 10.3 minutos apróx. La recarga para el plan internacional es mucho más baja igual, con una media de 3 recargas y una desviación estándar de 0.75. Tiene un máximo de 5 recargas para este plan internacional.

Luego, es importante que porcentaje de usuarios en esta base de datos abandonó la empresa o no, a continuación, se presenta un gráfico de torta con lo mencionado:

Figura 2: Gráfico de torta binario que indica si un usuario sigue en la empresa o no.

Se destaca que la gran mayoría de los usuarios siguen en la empresa y solo un 14.5% la abandonaron. Debido lo anterior, puede ocurrir que la diferencia entre la cantidad de usuarios que siguen o no en la empresa genere una sobreestimación hacia los usuarios que efectivamente siguen en la empresa, si es que la variabilidad de estas 2 variables no sea proporcional entre sí.

En consiguiente, nace la duda de la cantidad de usuarios por cada estado de Estados Unidos, ya que según el estado puede variar la señal que tiene la empresa para sus diferentes servicios y por ende, que en algunos estados tenga mayor influencia en el abandono de la empresa o no. Se presenta un gráfico de barras para describir lo anterior:

Figura 3: Gráfico de barras que indica la cantidad de usuarios por cada estado de Estados Unidos.

Se logra apreciar el estado de Virginia Occidental, tiene una diferencia de al menos 20 usuarios más que los otros y además, es el estado con más usuarios de la compañía "Orange Telecom", con un total de 106 usuarios. Por otro lado, el estado con menos usuarios respecto a los otros sería el estado de California, con un total de 34 usuarios registrados en la empresa.

En cuanto a los demás estados, se logra apreciar que varían todos, pero no por mucha diferencia, lo que representa una representatividad apróx. de usuarios en cada estado (sin contar aún la cantidad de habitantes que tiene cada uno).

Por otra parte, es relevante compara la variable "total de llamadas a servicio al cliente" con la variable churn que indica si el cliente abandonó la empresa o si sigue. Uno esperaría que según aumente la cantidad de llamadas al servicio al cliente, también aumente la cantidad de usuarios que abandonan la empresa, pues si un usuario llama al servicio al cliente es porque seguramente tiene problemas con el servicio. Se presenta el gráfico:

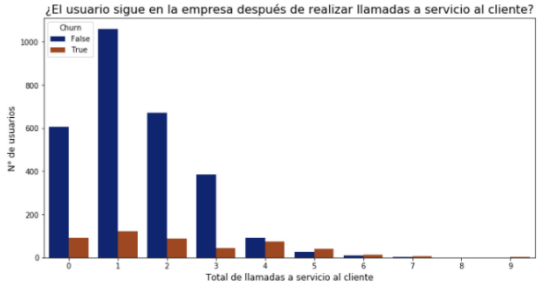

Figura 4: Gráfico de barras doble con las variables cantidad de llamadas al servicio al cliente y si el usuario abandona la empresa o no.

Se logra apreciar que al menos entre 0 y 3 llamadas que el usuario realiza a servicio al cliente, mayoritariamente decide quedarse en la empresa. Sin embargo, a medida que aumente de esta cifra las llamadas a servicio al cliente (entre 4 y 9) se aprecia que la proporción de usuarios que abandonan la empresa va aumentando, incluso superando a la proporción de usuarios que siguen en la empresa como se ve en el gráfico con 5 llamadas a servicio al cliente.

Por último, se observa la correlación entre las variables para poder estudiar posibles problemas de multicolinealidad con un pairplot:

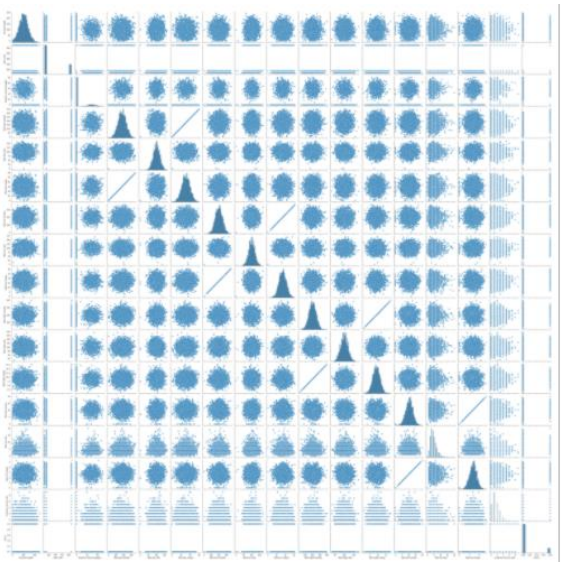

Figura 5: Pairplot de correlación entre las distintas variables de la base de datos.

Se logra apreciar que solamente las variables totales charge y total minutes (en sus 4 diferentes categorías) están correlacionadas, lo que es lógico. Las demás variables presentan que en general no hay multicolinealidad.

C. Estudios de Simulación

Se presenta las distintas métricas de desempeño obtenidas de cada clasificador en el conjunto de prueba, luego de entrenar el modelo:

TABLA 2
MÉTRICAS DE DESEMPEÑO PARA CADA CLASIFICADOR

Clasificador	Accuracy	Recall	Precisión	Especificidad	Medida F
LDA	0.865	0.282759	0.569444	0.984795	0.377880
QDA	0.883	0.572414	0.601449	0.935673	0.586572
Arbol de decisión	0.916	0.551724	0.808081	0.977778	0.655738
Random Forest	0.950	0.744828	0.892562	0.989474	0.812030
Regresión Logís	0.873	0.241379	0.673077	0.980117	0.355330
SVC Lineal	0.855	0.000000	0.000000	1.000000	0.000000
SVC Radio basal	0.889	0.627586	0.814885	0.933333	0.621160
Perceptrón	0.780	0.420690	0.309645	0.840936	0.356725
Perceptrón Multicapa	0.911	0.620690	0.725806	0.960234	0.669145
K-neighbors	0.899	0.413793	0.789474	0.981287	0.542986
Gradient Boosting	0.957	0.793103	0.898438	0.984795	0.842491
XGBoost	0.953	0.793103	0.871212	0.980117	0.830325

Según los resultados obtenidos, se puede apreciar que según la métrica de desempeño Accuracy los 3 clasificadores que

tuvieron un mejor porcentaje del total de elementos clasificados correctamente del total de clasificaciones, fueron el Random Forest, Gradient Boosting y el XGBoost, teniendo un gran porcentaje lo que indica un buen ajuste. Mientras los que tuvieron peor evaluación son el Perceptrón, SVC Lineal y el LDA.

En consiguiente, según el recall que mide la capacidad del clasificador de poder detectar correctamente los casos positivos, los mejores clasificadores fueron nuevamente los mismos, el Random Forest, el Gradient Boosting y el XGBoost, mientras que los peores evaluados en este ítem son el LDA, Regresión Logística (estos dos por muchísimo, solo obtuvieron un porcentaje menor al 30%) y también el SVC Lineal.

Según la precisión, que mide la probabilidad de que si una instancia x es clasificada en la clase c, la instancia realmente pertenece a esa clase, los mejores evaluados son nuevamente el Random Forest, Gradient Boosting y el XGBoost, mientras que los peores evaluados son nuevamente el LDA, el Perceptrón (por mucho, solo un 30%) y el QDA.

Según la especificidad, que mide la capacidad del clasificador de poder detectar correctamente los casos negativos, los mejores evaluados fueron el SVC Lineal (tuvo desempeño perfecto), Gradient Boosting, el XGBoost y el Random Forest nuevamente, sin embargo, en general todos tuvieron buenos resultados en esta métrica.

Finalizando, según la medida F que indica la media armónica entre la precisión y el recall, los que tuvieron mejor evaluación fueron nuevamente el Random Forest, Gradient Boosting y el XGBoost, mientras que los peores evaluados son el LDA, SVC Lineal y la Regresión Logística.

En consiguiente, se realizó ACP obteniendo el siguiente resultado:

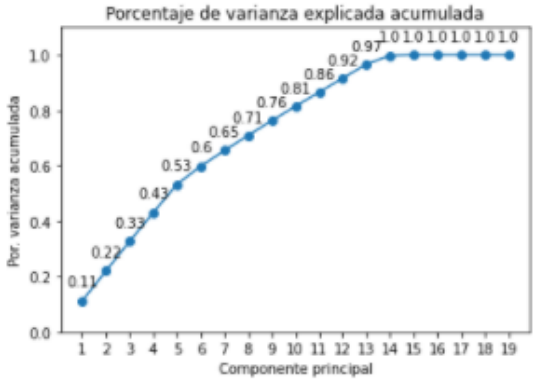

Figura 6: Porcentaje de varianza acumulada explicada de cada variable.

Donde se logra apreciar que desde la variable 8, las otras variables no explican un porcentaje de variabilidad alto. Lo anterior es lógico, pues antes con el análisis de correlación se identificó que las variables de totales minutos, recarga y llamadas en los diferentes horarios tienen una conrrelación cercana a 1, por tanto, se decide dejar solo las primeras 8 variables y, además, la de llamadas al servicio al cliente. Luego de aplicar los distintos clasificadores, se obtuvo lo siguiente:

TABLA 3
TABLAS DE ESTADÍSTICAS DESCRIPTIVAS BÁSICAS POSTERIOR A ACP

Clasificador	Accuracy	Recall	Precisión	Especificidad	Medida F
LDA	0.869	0.275862	0.606061	0.969591	0.379147
QDA	0.867	0.517241	0.543478	0.926316	0.530035
Arbol de decisión	0.887	0.544828	0.626984	0.945029	0.583026
Random Forest	0.887	0.482759	0.648148	0.955556	0.553360
Regresión Logís	0.868	0.193103	0.651163	0.982456	0.297872
SVC Lineal	0.855	0.000000	0.000000	1.000000	0.000000
SVC Radio basal	0.871	0.482759	0.564516	0.936842	0.520446
Perceptrón	0.850	0.020690	0.272727	0.990643	0.038462
erceptrón Multicapa	0.883	0.468966	0.629630	0.953216	0.537549
K-neighbors	0.870	0.372414	0.580645	0.954386	0.453782
Gradient Boosting	0.899	0.503448	0.715686	0.966082	0.591093
XGBoost	0.892	0.551724	0.650407	0.949708	0.597015

Se aprecia que las distintas evaluaciones disminuyeron en los clasificadores, por tanto, luego de realizar ACP se perdió un poco de confiabilidad en los resultados. Sin embargo, nuevamente los mejores clasificadores según las distintas métricas fue el Random Forest, Gradient Boosting y XGBoost, destacando por un poco encima del resto el Gradient Boosting.

IV. CONCLUSIONES

En conclusión, se puede deducir que en general el modelo Gradient Boosting fue por poco el que mejor se adaptó a la base de datos para poder clasificar, debido a que en la mayoría de las métricas de desempeño fue el mejor evaluado. Seguido por el XGBoost y en tercer lugar el Random Forest. Se destaca que el XGBoost al ser una versión mejorada del Gradient Boosting, no pudo obtener mejores métricas de desempeño que el Gradient Boosting, lo que puede ser contraproducente.

Además, se obtuvo clasificadores que son bastante buenos para detectar correctamente los casos verdaderos positivos, como el Random Forest y el árbol de decisión, pero estos no son tan buenos clasificadores para detectar los casos verdaderos negativos, lo que produce diferencias importantes en las diferentes métricas.

Por otro lado, clasificadores como el LDA, SVC Radio Basal y el Perceptrón Multicapa si tuvieron buen

desempeño para detectar verdaderos negativos, sin embargo, no tan buen desempeño para detectar los verdaderos positivos.

Entonces, si se tuviera que elegir un clasificador para un estudio sobre esta base de datos y las variables elegidas, se debería tomar como clasificador preferente el Gradient Boosting.

REFERENCIAS

Amat, Joaquín. (2020). Máquinas de Vector Soporte (SVM) con Python. https://www.cienciadedatos.net/documentos/py24-svm-python.html

Amat, Joaquín. (2020). Gradient Boosting con Python. https://www.cienciadedatos.net/documentos/py09_gradient_boosting_python.html

Barrientos, F. & Rios, S. (2013). Aplicación de Minería de Datos para Predecir Fuga de Clientes en la Industria de las Telecomunicaciones. Revista Ingeniería de Sistemas, 27, pp. 73-107. http://www.dii.uchile.cl/~ris/RIS2013/rios.pdf

Borja-Robalino, Ricardo & et al. (2020). Estandarización de métricas de rendimiento para clasificadores Machine y Deep Learning. *Revista Ibérica de Sistemas e Tecnologías de Informação, 30, 184-196.*

https://www.researchgate.net/profile/Antonio-Monleon-Getino/publication/342009715_Estandarizacion_de_metricas_de_rendimiento_para_clasificadores_Machine_y_Deep_Learning/links/5ede3d0392851cf13869078e/Estandarizacion-de-metricas-de-rendimiento-para-clasificadores-Machine-y-Deep-Learning.pdf

Gavilán, I. (2017). Clasificadores: El encuentro entre Data Science, Machine Learning y Redes Neuronales. https://ignaciogavilan.com/clasificadores-el-encuentro-entre-data-science-machine-learning-y-redes-neuronales/

Guarneros-Rivera, Manuel & et al. (2017). Reconocimiento de patrones en gráficos de control utilizando una red neuronal. *Revista de Tecnología Informática, 1(2), 1-8*. https://www.ecorfan.org/spain/researchjournals/Tecnologia_Informatica/vol1num2/Revista_de_Tecnologia_Informatica_V1_N2.pdf#page=16

Mejías César, Yuleidys & et al. (2013). Funciones de transferencia en el perceptrón multicapa: efecto de su combinación en entrenamiento local y distribuido. *Revista Cubana de Informática Médica, 5(2), 186-199.*

http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1684-18592013000200009&lng=es&tlng=es

Miranda, J. & et al. (2005). Prediccion de Fugas de Clientes para ´ una Institución Financiera mediante Support Vector Machines. Revista Ingeniería de Sistemas, 19, pp. 49-68. http://www.dii.uchile.cl/ris/RISXIX/RISXIXpaper4.pdf

Mosquera, R. & et al. (2016). Metodología para la Predicción del Grado de Riesgo Psicosocial en Docentes de Colegios Colombianos utilizando Técnicas de Minería de Datos. *Información tecnológica, 27(6), pp. 259-272.*
https://scielo.conicyt.cl/pdf/infotec/v27n6/art26.pdf

Jélvez, A. & et al. (2014). Modelo predictivo de fuga de clientes utilizando minería de datos para una empresa de telecomunicaciones en Chile. *Universidad, Ciencia y Tecnología, 18(72), pp. 100-109.*
http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S1316-48212014000300004

Timarán-Pereira, S. & et al. (2016). El proceso de descubrimiento de conocimiento en bases de datos. *Descubrimiento de patrones de desempeño académico con árboles de decisión en las competencias genéricas de la formación profesional, pp. 63-86.*
https://ediciones.ucc.edu.co/index.php/ucc/catalog/download/36/40/2301?inline=1#:~:text=El%20Descubrimiento%20de%20conocimiento%20en,que%20el%20usuario%20los%20analice.