INSTITUTO TECNOLÓGICO DE BUENOS AIRES – ITBA

ESCUELA DE (INGENIERÍA Y TECNOLOGÍA – INGENIERÍA Y GESTIÓN - POSTGRADO)

Modelo predictivo para disminución de la tasa de portabilidades numérica a través de redes sociales

AUTOR: Hernández Santiago Andres (Leg. Nº 104054)

TUTOR: Dr. Marcela Riccillo

TRABAJO FINAL PRESENTADO PARA LA OBTENCIÓN DEL TÍTULO DE ESPECIALIZACIÓN EN CIENCIAS DE DATOS

BUENOS AIRES

PRIMER CUATRIMESTRE, 2019

Índice General

Capítulo 1

Introducción

En Argentina unos de los mayores problemas que presentan las compañías de telefonía móvil es la pérdida de clientes, en el año 2012 a través de la ley Resolución 67/2011 Régimen de Portabilidad Numérica [1] les permitió a los usuarios de telefonía móvil cambiar de una compañía a otra manteniendo su número, en un plazo no mayor a 10 días. Esto provocó una guerra de precios y ofertas agresivas por parte de las compañías de telefonía celular, que, sumado a otros problemas existentes, provoco que en 2018 las cifras de portabilidad llegaran a triplicarse según datos correspondientes a ENACOM [2].

En el año 2016**,** las reglas en el mercado argentino fueron cambiando, se permitió el ingreso de OMV (Operadoras Virtuales) al mercado según **Resolución 6033/2016 ENTE NACIONAL DE COMUNICACIONES**, como así también las operadoras de telefonía celular podrían brindar servicios 4 play (Móvil, Banda Ancha, TV, Telefonía Fija) **Decreto 1340/2016**. Estos cambios generaron un mercado más competitivo.

Dado este contexto, resulta necesario para las compañías de telefonía celular, realizar un análisis de sus clientes en base a los que más expuestos están a cambiarse de operador. Por tanto, resulta necesario desarrollar herramientas de predicción que permitan estimar cuales son aquellos clientes que están en riesgo de abandonar la compañía como así también entender cuál es el motivo de su disconformidad.

1.1 Marco Conceptual

1.1.1 Portabilidad Numérica

En la publicación realizada por ENACOM en su resolución sobre portabilidad numérica [2] reglamenta la portabilidad, en la misma se define a la portabilidad como la posibilidad que tienen los usuarios de telefonía móvil de cambiar de compañía conservando su número telefónico, el proceso de portabilidad numérica no podrá ser mayor a DIEZ (10) días hábiles. La Autoridad Regulatoria podrá modificar el plazo en base a la experiencia adquirida en la práctica del Proceso de Portabilidad Numérica y a los avances tecnológicos.

La portabilidad numérica para celulares se reglamentó en 2000 y entró en vigencia el día jueves 1º de marzo de 2012 [17].

Con esta facilidad que disponen los clientes, la portabilidad numérica creció un 338% desde que fue creada dicha ley según datos correspondientes a ENACOM [2], el mismo surge del cálculo del cuadro 1 donde en 01/2014 las portabilidades eran igual a 30.223 y en 07/2018 132.419

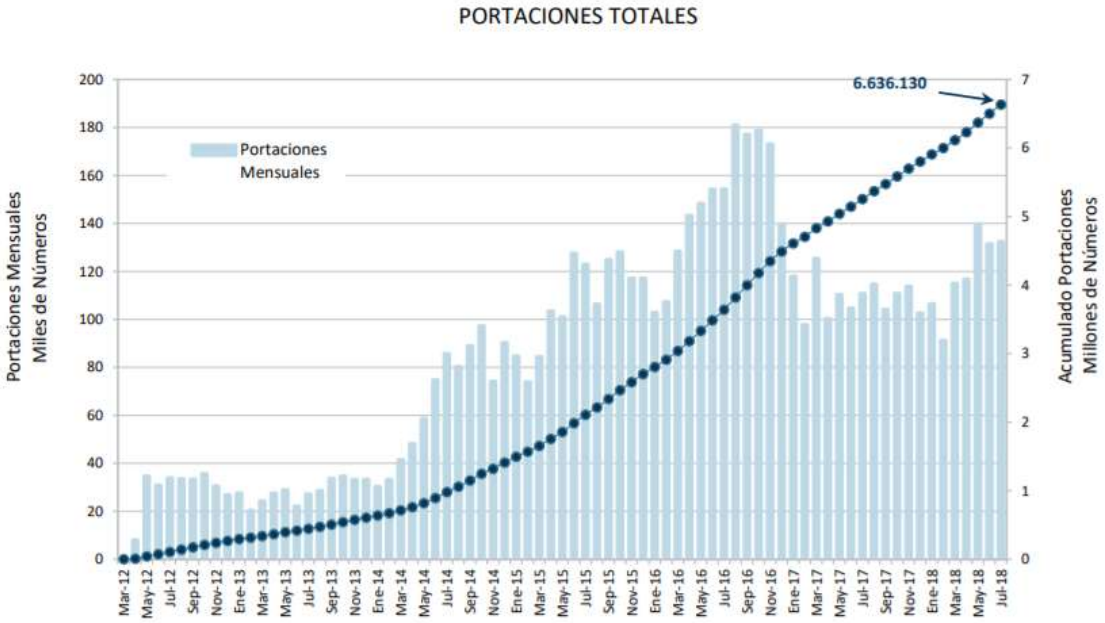

Grafico 1.1 Cantidad portabilidad numérica Argentina 2012 a 2018 ENACOM [2]

AÑO 2014			AÑO 2015			AÑO 2016			AÑO 2017			AÑO 2018		
Mes	Portaciones	Acumulado	Mes	Portaciones	Acumulado	Mes	Portaciones	Acumulado	Mes	Portaciones	Acumulado	Mes	Portaciones	Acumulado
Ene	30.223	638.478	Ene	84.687	1.495.663	Ene	102.891	2.804.973	Ene	117.961	4.608.280	Ene	106.465	5.909.594
Feb	33.185	671.663	Feb	73.881	1.569.544	Feb	107.273	2.912.246	Feb	97.738	4.706.018	Feb	91.147	6.000.741
Mar	41.372	713.035	Mar	84.411	1.653.955	Mar	128.411	3.040.657	Mar	125.312	4.831.330	Mar	114.972	6.115.713
Abr	48.043	761.078	Abr	103.422	1.757.377	Abr	143.274	3.183.931	Abr	100.241	4.931.571	Abr	116.763	6.232.476
May	58.527	819.605	May	100.936	1.858.313	May	148.327	3.332.258	May	110.304	5.041.875	May	139.781	6.372.257
Jun	74.669	894.274	Jun	127.486	1.985.799	Jun	154.196	3.486.454	Jun	104.675	5.146.550	Jun	131.454	6.503.711
Jul	85.713	979.987	Jul	122.893	2.108.692	Jul	154.234	3.640.688	Jul	110.699	5.257.249	Jul	132.419	6.636.130
Ago	80.264	1.060.251	Ago	106.316	2.215.008	Ago	180.968	3.821.656	Ago	114.627	5.371.876	Ago	-	-
Sep	88.994	1.149.245	Sep	124.779	2.339.787	Sep	177.072	3.998.728	Sep	104.109	5.475.985	Sep	-	-
Oct	97.254	1.246.499	Oct	128.001	2.467.788	Oct	178.771	4.177.499	Oct	110.786	5.586.771	Oct	-	-
Nov	74.132	1.320.631	Nov	117.154	2.584.942	Nov	173.043	4.350.542	Nov	113.740	5.700.511	Nov	-	-
Dic	90.345	1.410.976	Dic	117.140	2.702.082	Dic	139.777	4.490.319	Dic	102.618	5.803.129	Dic	-	-

Cuadro 1.1: Cantidad portabilidad numérica Argentina 2012 a 2018 ENACOM [2]

Como se puede observar en el cuadro 1.1 la cantidad de portabilidad se triplicaron desde 2014 a 2018. Llegando a acumular un total de 6.636.130 desde 2012.

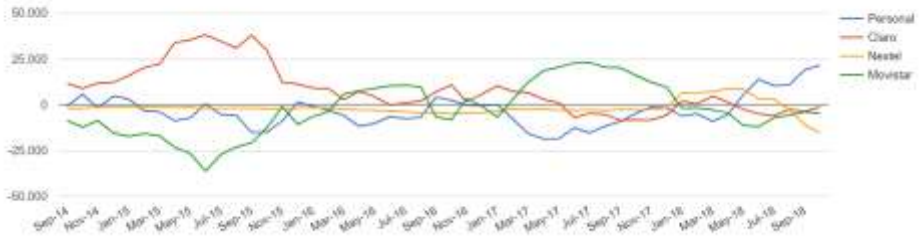

Grafico 1.2 Portabilidad numérica móvil: Portaciones netas mensuales por operador [2]

En el grafico 1.2 vemos como los operadores de telefonía móvil van cambiando sus estrategias de captación de clientes, donde el liderazgo en la portabilidad va cambiando. También se observa en el grafico que la pérdida de clientes de un operador representa un alta en otro operador.

1.1.2 ¿Qué es el churn?

En el trabajo realizado por Arnejo Calviño [3] define al Churn o tasa de cancelación como el porcentaje de clientes o suscriptores que dejan de utilizar los servicios que ofrece una empresa durante un período de tiempo determinado.

$$\text{Churn} = \frac{(\text{N° de clientes al principio} - \text{N° de clientes al final})}{\text{N° de clientes al comienzo del periodo}} \times 100$$

El hecho de que un cliente abandone la compañía presenta un efecto temporal ya que la decisión de baja es un estado transitorio. Lo más probable es que este cliente abandone una compañía para darse de alta en otra, operación que conocemos como portabilidad.

1.1.3 Principales razones de cambio de compañía

Los principales motivos por lo cual los clientes cambian de compañía según el trabajo realizado por Galván et al. [5] son los siguientes:

- Mala señal

- Cobertura

- Aumento de precios

- Falta de políticas de retención por parte de las compañías

-Falta de inversión

-Ofertas convergentes

-Redes de contactos también llamadas comunidades (Es la posibilidad de comunicarse en forma gratuita entre los usuarios de la misma compañía).

1.1.4 Redes Sociales y portabilidad

Una red social es una estructura social compuesta por un conjunto de actores (tales como individuos u organizaciones) que están relacionados de acuerdo a algún criterio (relación profesional, amistad, parentesco, etc.). Las redes sociales pueden influenciar distintos aspectos de una persona como por ejemplo contratar un servicio, comprar un producto o el hecho de abandonar una compañía, todo esto mediante las referencias o el conocido efecto “boca a boca”.

Dentro de una red existen lazos que unen a dos o más actores y estos pueden ser débiles o fuertes basados en las interacciones que tiene dichos agentes.

Existen estudios tal como el realizado por Pérez Villanueva et al. [4] donde miden a las telecomunicaciones como redes sociales, se estudia a las redes de contactos como

una red social, donde la red social se puede medir mediante la cantidad de llamadas, mensajes de textos o tráfico de información entre los distintos clientes, los que tienen mayor actividad serían los líderes y los de menor trafico serían los seguidores. En la actualidad este tipo de tráfico está tendiendo a disminuir, dejando este modelo como obsoleto.

1.1.5 Soluciones actuales

Las compañías de telefonía móvil, suelen realizar modelos predictivos para poder identificar cuáles son aquellos clientes más propensos a darse de baja del servicio. Se debe tener en cuenta unas series de variables antes de realizar dicho modelo como sistema de pago, factores demográficos, segmento, NSE, todo lo referido al riesgo crediticio. Otras variables a tener en cuenta serian: Cantidad de minutos en llamadas (Realizado y Recibidos), SMS y cantidad de MB consumidos podemos nombrar en caso del trabajo online realizado por Chiang et al.[6] que utiliza estos modelos . También se podría tener en cuenta las páginas web donde navegan dichos clientes como si recibieron algún llamado de Call Center de la competencia.

1.1.6 Problemas que generan las soluciones actuales

Entre los principales problemas podemos mencionar que las portabilidades van a mayor ritmo que las predicciones, la confidencialidad en los
datos, leyes que regulan la usabilidad de datos, accesos a datos, entre otros.

1.2 Estado del arte

En el trabajo realizado por Galván et al. [5] la solución propuesta es obtener en tiempo real los comentarios dejados en las fanpages oficiales de Facebook de las operadoras móviles, y analizar el sentimiento expresado para poder predecir si un cliente tiene intención de realizar una portabilidad hacia la competencia, donde utiliza otros modelos de predicción distintos al presente trabajo. Este trabajo tiene un gran limitante, ya que su análisis solo llega a la clasificación de comentarios en positivos o negativos, en el presente trabajo se intentara la apertura de los comentarios negativos para dar mayor precisión al análisis.

En el modelo de predicción de fuga de clientes realizado Pérez Villanueva et al. [4] los resultados muestran que con el modelo árbol de decisión se obtiene la mayor cantidad de aciertos, lo cual solo logra obtener un 41% de exactitud en la predicción y un sensitivity del 73%, lo que indica que la predicción del modelo no es buena.

En el trabajo online Predicting customer churn with scikit-learn realizado por Chiang et al.[6] utiliza tres algoritmos para predecir la tasa de Chrun de una telco. Este trabajo utiliza variables como ser cantidad de llamadas o duración de las mismas donde como fue mencionado con anterioridad estos métodos de predicción detectan la fuga cuando ya fue realizada la misma.

1.3 Definición del problema

El sistema de portabilidad numérica permite actualmente en Argentina que un cliente cambie de empresa de telefonía celular manteniendo su número en un plazo no mayor a 10 días. Esta facilidad en la rotación entre compañías de servicios, sumado a la disconformidad de los usuarios debido a la calidad en la señal, los precios, inconvenientes en los pagos, y las faltas de políticas de retención genera grandes fugas de clientes en estas empresas. Asimismo, la falta de detección anticipada de los clientes propensos a realizar la portabilidad numérica limita a las empresas a definir estrategias de comercialización y de servicios que mejoren los niveles de retención de sus clientes. Esto se volvió un problema para las empresas y lo que se busca es identificar cual es la problemática que lleva a los clientes a realizar la portabilidad numérica.

1.4 Justificación del estudio

La investigación se llevará a cabo sobre las opiniones en Twitter sobre compañías de telefonía celular, la idea no solo es poder mejorar la tasa de Churn de la compañía si no también la experiencia de los clientes de la misma, para poder entender los motivos de disconformidad de los mismos.

Lo que intentamos cubrir en el presente trabajo, en relación a otros autores, más allá de los comentarios positivos o negativos, es segmentar opiniones en grupos homogéneos

de acuerdos a su disconformidad su compañía actual. A su vez aislar a aquellos que tienen mayor probabilidad de portar de compañía y poder entender su problemática actual. Esta probabilidad debe entenderse como aquel cliente que mencionó su intención de cambiar de compañía.

1.5 Alcances del trabajo y limitaciones

Dentro de los alcances y limitaciones podemos mencionar:

-Solo se podrá realizar predicciones sobre aquellas personas que realicen comentarios en alguna red social.

-Algunas redes sociales como ser Twitter limitar su extracción de datos a través de su API, en este ejemplo el límite diario de descarga es de: 600 tweets diarios.

-El análisis solo corresponde a aquellos clientes que sean personas físicas.

1.6 Hipótesis

Logrando entender cuál es la disconformidad que posee un cliente a través de su comentario en Twitter, se podrá lograr disminuir la tasa de churn de una compañía de telefonía móvil.

1.7 Objetivos

1.7.1 Objetivos generales

Generar un modelo de predicción que permita a través de los comentarios de una red social segmentar a los cometarios de los usuarios de las compañías de telefonía celular. De esta forma se podrá ofrecer una solución acorde a cada de las necesidades del cliente tratando de disminuir las bajas de la compañía.

1.7.2 Objetivos específicos

*Predecir comentarios positivos o negativos.

*Elaborar una clasificación de comentarios negativos, este servirá para agrupar a los clientes.

*Clasificar los comentarios negativos según su problema.

*Validar el modelo creado y definir el porcentaje de coincidencias.

*Categorizar a los comentarios de la red social.

Capítulo 2

Metodología-Técnicas/Herramientas

2.1.1 Extracción de Twitter

La primera técnica que se desarrolla corresponde a minería de las redes social, esto será útil para poder extraer los datos con el fin de desarrollar la predicción.

El primer paso es conseguir la clave de la API de Twitter en https://apps.twitter.com/.

Una vez obtenida esta clave se precederá a la extracción de tuits, para ello se decidió utilizar una herramienta provista por la nube de google drive a través del complemento Twitter Archiver tal como lo desarrolla Agarwal et al.[16]. Esta búsqueda es en tiempo real y no tendremos la limitación de límite de descarga como tampoco de los 140 caracteres. Esta se repetirá en forma automática y guardará automáticamente nuevos tuits una vez por hora, de esta forma nos evitamos correr códigos de forma manual.

Al principio se había decidido utilizar como referencia a Russell et al. [9], donde a través de una notebook de jupyter y con lenguaje de programación Python se procedió a la extracción de Twitter en forma Streaming. Con esta técnica nos encontramos con limitación al alcance, en las cuentas que no son empresariales o Premium están limitadas a descargar 140 caracteres y a la vez no es posible extraer información importante sobre los usuarios, como ser fecha, nombre de usuario, ID, localidad, sistema

operativo, etc. Otro problema que se genero fue que Python al segundo día de procesamiento cortaba el proceso y deberíamos correrlo nuevamente.

Por este motivo de decidió la utilización de la herramienta provista por google drive en vez de Python.

De esta forma nos evitamos de la recolección de información.

El formato de Tuits provisto por Python cuando superaba los 140 caracteres es el siguiente:

"¿Qué estás dispuesto a hacer x la selección? Contanos tu promesa, menciona @movistararg + #PromesasMovistar y participar por un….Link"

El cambio el mismo Tuits cuando se extrae con Google drive tiene el siguiente formato:

Date	Screen Name	Full Name	Tweet Text	Tweet ID	App	User Since	Location
6/19/2018			¿Qué estás dispuesto a hacer x la selección? Contanos tu promesa, menciona @movistararg + #PromesasMovistar y participar por un Samsung Galaxy s9+. Las más creativas las vamos a compartir en el programa del mundial y el premio será para una de ellas. ?????????????		Twitter for Android		

2.1.2 Pre Procesamiento de texto

En esta segunda fase, una vez que se obtuvieron los tuits suficientes, se procederá al armado del dataset que nos permitirá realizar los modelos predictivos. Para este caso se utilizará R que es un entorno integrado para manipulación de datos, cálculos estadísticos y gráficos.

Usaremos librerías como tm que es text mining que es utilizada para minería de texto en R, también usaremos SnowballC que sirve para llevar las palabras a su raíz, o sea sin conjugación verbal.

Una vez cargados todos los tuits previamente clasificados (Mas adelante se explicarán dicha clasificación) se procederá a convertir estos datos no estructurados en datos estructurados. Lo primero que haremos será crear el Corpus, posteriormente se procederá a extraer los Stop Words (son las palabras más frecuentes de la lengua, Tales como: "de" "la" “que" "el" “en” “y" "a" "los" "del" "se") se eliminaran espacios, puntuación, mayúsculas, simbología que no esté relacionado con los Tuits, stemming: que es recortar la palabra a su raíz (Sacar el tiempo verbal).

A continuación se mostrara un ejemplo de entrada de Twitter a la base de datos.

Las filas van a contener cada uno de los tuits y las columnas o sea las variables van a ser todas las palabras posibles que puedan aparecer en nuestros Tuits, cada vez que la palabra aparece en el texto se la condicionar con un 1, si no aparece lleva un 0, si la misma aparece más de una vez se indica con un 2 por ejemplo.

Tweet	Sentimiento	Palabra 1	Palabra 2	Palabra 2	…	Palabra N
1	0	1	0	0	…	…
2	0	0	1	0	…	…
3	1	1	0	1	…	…
…	…	…	…	…	…	…
N	…	…	…	…	…	…

Cuadro 2.1: Entrada Twitter a data set

Luego de realizado el pre procesamiento se va a crear la matriz antes detallada, la misma se va a crear con la función DocumentTermMatrix.

2.1.3 Desbalanceo de clases

El desbalanceo de clases es una característica de la muestra que se produce cuando una o más clases (clases minoritarias) se encuentran representadas en menor medida que otras (clases mayoritarias).Esta cuestión supone un problema en el ámbito de la clasificación de nuevos datos ya que puede derivar en un deterioro notable en la ciencia de nuestro clasificador. En particular, desempeñara una buena labor sobre la clase mayoritaria en detrimento de la minoritaria ya que detecta una mayor presencia en la muestra del primer grupo decantando la balanza sobre la clase mayoritaria a la hora de clasificar.

Basándonos en el trabajo realizado por Arnejo Calviño et al. [3] existen dos metodologías o maneras de afrontar el desbalanceo de datos:

1. Técnicas de re muestreo: se basan en modificar la distribución inicial de los datos para balancear las clases. Algunas de las más importantes:

Oversampling: Consiste en modificar la distribución de los datos incrementando el número de casos de la clase minoritaria.

Undersampling: Consiste en modificar la distribución de los datos reduciendo el numero

De casos de la clase mayoritaria.

Algoritmos híbridos: Combinaremos las técnicas de undersampling y oversampling.

2. Modificación de algoritmos: consiste en variar los algoritmos existentes para mejorar la predicción.

Los dos enfoques son independientes entre sí y pueden combinarse para mejorar el rendimiento de cada uno.

La técnica que usaremos en nuestro trabajo será: Oversampling. Usaremos esta técnica ya que con técnicas como Undersampling existe el riesgo de eliminar elementos de la muestra potencialmente importantes en el proceso de clasificación,

A continuación, detallaremos esta técnica según el trabajo realizado por Arnejo Calviño et al. [3] donde nos dice que consiste en balancear la distribución de los datos añadiendo ejemplos de la clase minoritaria, es decir, del grupo de clientes que se dan de baja. Además de aumentar el tiempo de procesado de los datos, el principal problema de este método tiene lugar cuando generamos ejemplos ruidosos que se producen al realizar un elevado número de réplicas de ciertas instancias, modificando en exceso la distribución de las bajas, y que puede derivar en un incremento desmesurado de las probabilidades de baja finales.

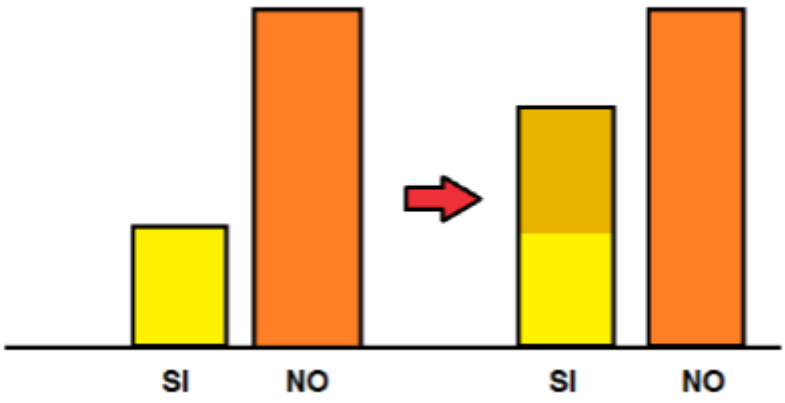

Grafico 2.1: Oversampling [3]

2.1.4 Entrenamiento del modelo

Se realizará la prueba con dos metodologías

1. El primer método divide los datos en los siguientes datasets:

    - Training. Este es el dataset que se utilizará para “entrenar” el modelo de predicción. Consta de una serie de variables (directas y derivadas a partir de las directas) y también de la variable respuesta. Es el dataset con el que aprende el modelo predictivo.
    - Testing. Este es el dataset que se utilizará para probar el modelo de predicción. Consta de la misma serie de variables desarrolladas en la tabla de training, pero no incluimos la variable respuesta en el conjunto (es la que vamos a predecir), aunque la conocemos y la utilizamos posteriormente para hallar la precisión del modelo. Sobre esta tabla se aplica el modelo.

.

2. El segundo método es Cross Validation:

    La validación cruzada o Cross-Validation intenta medir el sobreajuste (capacitación y predicción del mismo punto de datos) al tiempo que produce una predicción para cada conjunto de datos de observación. Esto se logra ocultando sistemáticamente diferentes subconjuntos de datos mientras se entrena un conjunto de modelos. Después del entrenamiento, cada modelo predice en el subconjunto que se le había ocultado, emulando múltiples divisiones de prueba de tren. Cuando se hace correctamente, cada observación tendrá una predicción correspondiente "justa”.

Luego se procederá a crear el data frame con la función as.data. Frame (as. matrix ()) y se procederá a introducir la variable de clasificación del sentimiento.

2.2 Algoritmos de predicción

A continuación, se desarrollará los algoritmos utilizados para la predicción de Tuits, en ella de medirá el porcentaje de aciertos que tiene el modelo.

2.2.1 Árbol de decisión para clasificación

En el trabajo realizado por Parra et al.[7] se define a los árboles de decisión para clasificación como un modelo surgido en el ámbito del aprendizaje automático (Machine Learning) y de la inteligencia artificial que partiendo de una base de datos, se crean diagramas de construcciones lógicas que nos ayudan a resolver problemas. A esta técnica también se la denomina segmentación jerárquica. Es una técnica explicativa y descomposicional que utiliza un proceso de división secuencial, iterativo y descendente que partiendo de una variable dependiente, forma grupos homogéneos definidos específicamente mediante combinaciones de variables independientes en las que se incluyen la totalidad de los casos recogidos en la muestra.

Se comienza con un nodo inicial, dividiendo la variable dependiente a partir de una partición de una variable independendiente que se escoge de modo tal que dé lugar a dos conjuntos homogéneos de datos (que maximizan la reducción en la impureza).

En los árboles de decisión se encuentran los siguientes componentes: nodos, ramas y hojas. Los nodos son las variables de entrada, las ramas representan los posibles valores de las variables de entrada y las hojas son los posibles valores de la variable de salida. Como primer elemento de un árbol de decisión tenemos el nodo raíz que va a representar la variable de mayor relevancia en el proceso de clasificación.

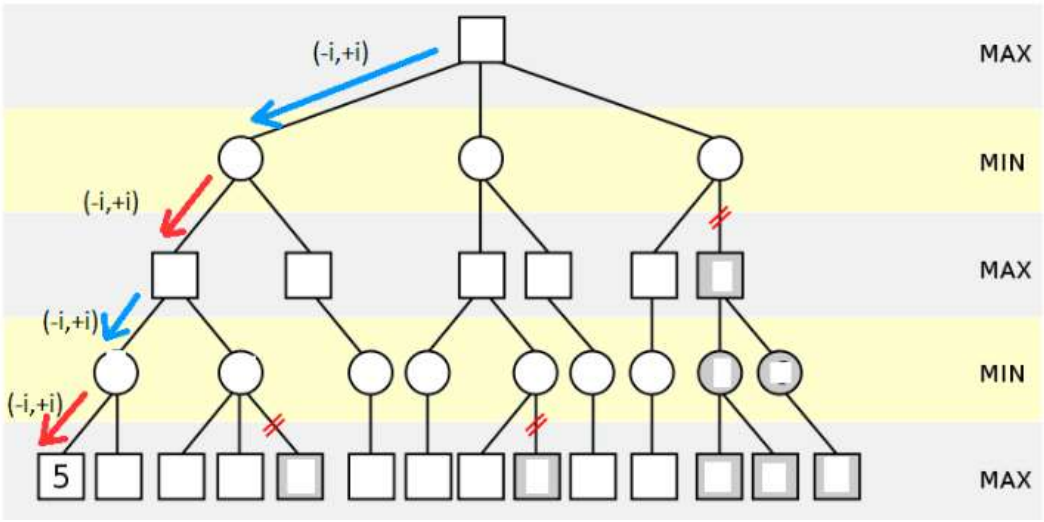

Grafico 2.2: Arboles de decisión para clasificación [7]

2.2.2. SVM (Support Vector Machines)

Tal como como lo describe Amat Rodrigo et al. [8] define a “Support Vector Machine" (SVM) como un algoritmo de aprendizaje automático supervisado que se puede utilizar para desafíos de clasificación o regresión. SVM es una implementación de los principios de minimización del riesgo de la estructura que busca minimizar un límite superior del error de generalización en lugar de minimizar el error empírico.

La Figura 1 muestra el hiperplano óptimo en SVM que separa dos conjuntos de datos, el vector cercano al hiperplano se llama vectores de soporte (SV). La precisión de un modelo SVM depende en gran medida de la selección de los parámetros del núcleo, ya que estos parámetros tienen un impacto significativo en el rendimiento del método kernel. El número de estos parámetros depende del margen que separa los conjuntos de datos.

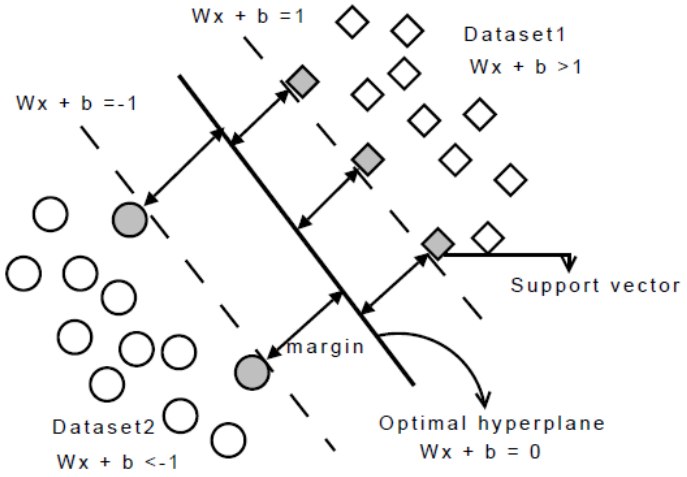

Grafico 2.3: Hiperplano optimo en SVM [8]

Los vectores de soporte son simplemente las coordenadas de la observación individual. Support Vector Machine es una frontera que mejor segrega las dos clases (hiperplano / línea). SVM declara que la mejor línea de separación va a ser la línea que divide las dos clases y al mismo tiempo maximiza la distancia de hiperplano de soporte.

El argumento kernel corresponde al núcleo que representa el producto escalar que queremos utilizar

El argumento cost determina la penalización que ponemos a los errores de clasificación.

Scale=FALSE se usa para usar los datos no estandarizados (por defecto, sí se estandarizan):

Diferencia entre Kernel que se utilizaran en el modelo [8]:

Kernel lineal

$$K(x, x') = x \cdot x' \quad K(x, x') = x \cdot x'$$

Si se emplea un Kernel lineal, el clasificador Support Vector Machine obtenido es equivalente al Support Vector Classifier.

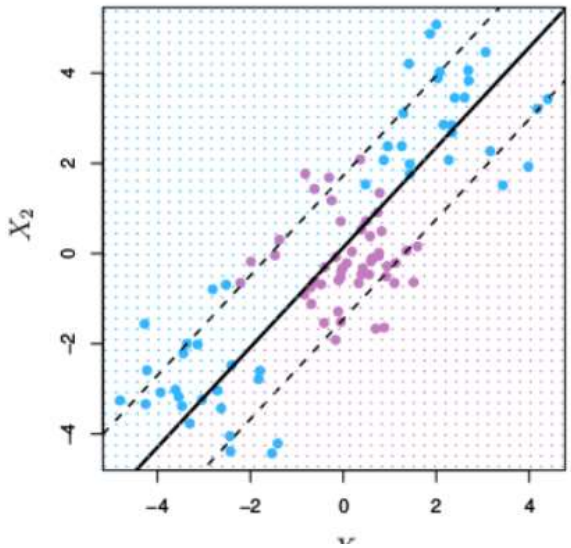

Grafico 2.4: Kernel Lineal [8]

Kernel polinómico

$$K(x, x') = (x \cdot x' + c)^d \quad K(x, x') = (x \cdot x' + c)^d$$

Cuando se emplea dd y c=0c=0, el resultado es el mismo que el de un kernel lineal. Si d>1, se generan límites de decisión no lineales, aumentando la no linealidad a medida que aumenta d. No suele ser recomendable emplear valores de dd mayores 5 por problemas de overfitting.

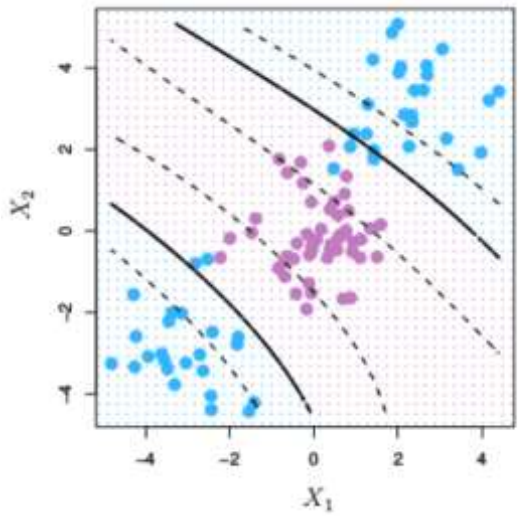

Grafico 2.5: Kernel polinómico [8]

2.3 Métricas de clasificación

Accuracy

La accuracy mide la frecuencia con la que el clasificador hace una predicción correcta. Es la relación entre el número de predicciones correctas y el número total de predicciones. Su fórmula general es la siguiente

$$\text{Accuracy} = \frac{\text{Predicciones correctas}}{\text{Individuos totales}}$$

Precisión y sensibilidad

La precisión y la sensibilidad son dos medidas diferentes, pero a menudo se utilizan conjuntamente. La precisión representa el porcentaje de casos positivos predichos correctamente. Por otro lado, la sensibilidad representa la capacidad para detectar una condición correctamente.

$$\text{Precisión} = \frac{\text{Verdaderos positivos}}{\text{N° de positivos predichos}}$$

$$\text{Sensibilidad} = \frac{\text{Verdaderos positivos}}{\text{Positivos reales}}$$

Especificidad

La especificidad se refiere a la capacidad del modelo para excluir correctamente una condición, en nuestro caso, corresponde a la capacidad para detectar un usuario realice un cometario negativo, por ejemplo. Matemáticamente, esto también puede ser escrito como:

$$\text{Especificidad} = \frac{\text{Verdaderos negativos predichos}}{\text{Negativos reales}}$$

Capítulo 3

3.1 Experimentación

3.1.1 Pre procesamiento de datos

- Extracción de Tuits

Como ya fue mencionado, la herramienta que usaremos para la extracción de Tuits será la provista por la nube de google drive. Esta búsqueda es en tiempo real y guardará automáticamente nuevos tuits una vez por hora.

Se aplicará como único filtro en "These #hashtags", para nuestro caso de @MovistarArg así que vamos a usarlo como base de una consulta de búsqueda para recuperar algunos tuits para un análisis posterior. Para este trabajo se extrajeron 2.500 tuits.

- Clasificación de Tuits:

En este paso se procederá a agregar las clases según los grupos definidos previamente, para esta primera etapa se realizaron la clasificación de 2.500 Tuits, la misma se tuvo

que hacer de forma manual, esta base nos servirá para realizar el aprendizaje automático y así poder predecir futuro Tuits

Clases para primer variable:

-Positivo

-Negativo

-Neutro

El criterio de clasificación para el caso de negativo, es cuando algún cliente manifiesta alguna disconformidad en algunos de los productos /Servicios que ofrece esta compañía. El resto se los clasificara como positivo o Neutro.

La segunda clasificación será para poder armar el catalogo en la disconformidad, para poder realizar esto se descompondrá a los Tuits negativos de la siguiente forma:

-Portabilidad

-Mala Señal/Cobertura/Problemas en servicio

-Problema de Facturación/pago/Aumento Precio

-Resto Negativo (Ejemplo: Problema con la venta de equipos Celulares/Smartphone)

El criterio de clasificación para esta clase será por ejemplo para Portabilidad, cuando algún cliente manifiesta directamente que va a realizar una portabilidad o que se lo sugiere a otros clientes.

3.1.2 Exploración de datos

Si inspeccionamos la matriz nos encontraremos con:

<<DocumentTermMatrix (documents: 2500, terms: 6024)>>
Non-/sparse entries: 30957/15029043
Sparsity: 100%
Maximal term length: 49
Weighting: term frequency (tf)

Donde nos dice que contiene 2500 twitter y 6024 términos o sea columnas o palabras

que no se repiten. La palabra más larga contiene 49 caracteres.

También podemos ver que las palabras con mayores frecuencias son:

[1]	"dndconsumidor"	"enacomargentina"	"hace"	"telefonicacomar"	"movistararg"	"contenidos"
[8]	"hola"	"gracias"	"necesito"	"movistar"	"funciona"	"datos"
[15]	"linea"	"factura"	"internet"	"respuesta"	"mes"	"servicio"
[22]	"personalar"	"nunca"	"voy"	"ver"	"ahora"	"reclamo"
[29]	"claroargentina"	"nea"	"casa"	"fono"	"puedo"	"semana"
[36]	"quiero"	"ustedes"	"pago"	"liganacional"	"liga"	"lorenzo"
[43]	"anda"	"problema"	"hacer"	"mas"	"meses"	"speedy"
[50]	"van"	"mal"	"argentina"	"promesasmovistar	"empresa"	"vez"

Cuadro 3.1: Palabras con frecuencia mayor a 50

- Nube de palabras: Previo a analizar los resultados obtenidos con cada uno de los métodos vamos a proceder a ver la nube de palabras para ver cuál son aquellas palabras más utilizadas en nuestra data set de aquellos comentarios negativos:

Grafico 3.1: Nube de palabras con Stop Word

A continuación se va a proceder a analizar la nube de palabras sin tener en cuenta aquella palabra que se repiten más de 50 veces, de esta forma se podrá visualizar aquella que aparecen con menor frecuencia.

Las palabras que poseen una frecuencia mayor a 50 son las siguientes:

Word	Frequencia
movistararg	1547
servicio	302
internet	238
hace	198
hola	161
movistar	110
telefonicacomar	104
gracias	76

Cuadro 3.2: Ranking de palabras más frecuentes

Las palabras con más frecuencia son movistararg que se repite 1547 veces, servicio 302 veces.

Se procede a armar nuevamente la nube de palabras sin tener en cuenta las palabras con mayor frecuencia:

Grafico 3.2: Nube de palabras sin palabras más frecuentes

Las palabras más destacadas: Son baja y factura, lo que se puede deducir es que los principales reclamos son intensión de baja como problemas en la factura, también se observa la presencia de las empresas de la competencia en la nube, estas serían: personalar, claroargentina, etc. lo que significa la intención de portabilidad a otra empresa que tienen los usuarios. Tal destacamos palabras de quejas de usuarios tales como reclamo, dndconsumidor (defensa del consumidor), respuesta, datos y pago.

3.2 Resultados

Para poder analizar los resultados se utilizaran dos modelos de datos distintos, el primer modelo se utilizara una única variable con 6 clases diferentes. Estas clases como ya vimos en puntos anteriores serán Positivos, Neutros y a diferencia de trabajos como Galván et al. [5], se realiza la apertura de los tuis negativos en portabilidad, problemas cobertura o señal, problemas de facturación y otros negativos.

En el segundo modelo la predicción se realizara en dos etapas, la primera será para predecir la primer variable que tendrá 3 clases (Positivos, negativos y neutros). Una vez realizada esta predicción son nos quedaremos con aquellos Tuits Negativos y se procederá a clasificar los mismos, para ellos utilizaremos 4 clases (Portabilidad, Problema técnico, Problema de facturación y otros problemas)

Al final del procedimiento se compararán dichos resultados y se verá cual es método que maximiza los resultados. Lo que intentamos comprobar cuál de estas metodologías maximiza los indicadores de las predicciones.

3.2.1 Modelo única variable

Como se dijo en el párrafo anterior se procederá a predecir todos los Tuits en una única variable con sus seis clases posibles. El primer modelo que usaremos para la dicha predicción será una Árbol de decisión para clasificación y luego analizarnos un modelo SVM.

La distribución de los 2.500 Twitter será:

Neutro	Otro problema	Portabilidad	Positivos	Problema facturación	Problema técnico	Total
706	468	222	230	243	631	2500

Cuadro 3.3: Distribución de Tuits única variable

Podemos observar que las clases minoritarias son Portabilidad, problema de facturación y positivos. En estas primeras pruebas no se aplica la técnica de Oversampling donde se observara los resultados sin tener en cuenta el desbalanceo de clase.

3.2.1.1 Arboles de decisión para clasificación

El primer algoritmo de predicción utilizado será Árbol de decisión para clasificación, según lo indicado en el trabajo realizado por [4] los resultados muestran que el mejor modelo es un Árbol de Decisión, el cual no logra obtener un alto valor de exactitud en la predicción.

Previamente graficaremos como se forma un árbol de decisión, para ello utilizaremos la librería rpart.plot en r. De esta manera podremos ver cuales con aquellas variables que presentan más importancia en cada una de las clasificaciones.

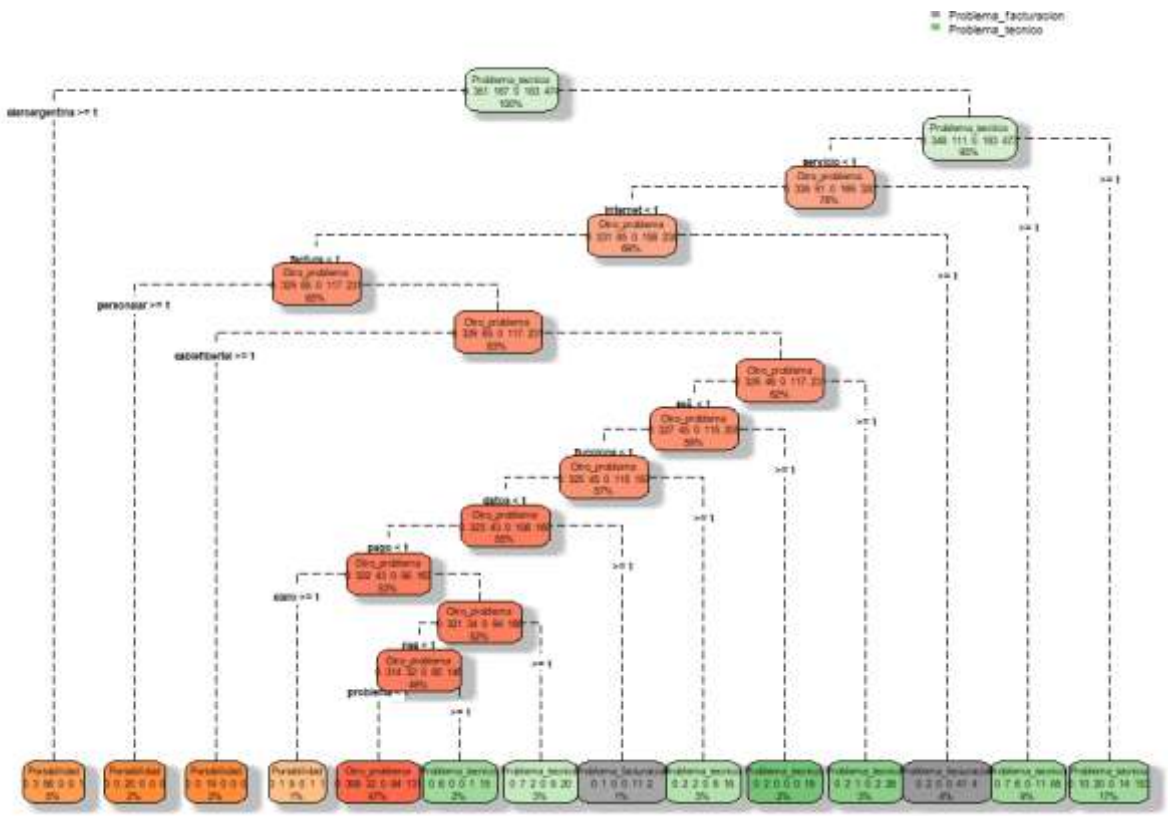

Grafico 3.3: Esquema árbol de decisión para clasificación

A continuación se expondrán los resultados obtenidos, utilizando para Train el 75% del data set y para test el 25% restante. Todos los modelos tendrán la misma distribución.

Para poder analizar los resultados obtenidos en el árbol utilizáramos una matriz de confusión y luego se procederá a ver todas las métricas de clasificación, esta nos servirá para realizar la comparativa con los distintos resultados obtenidos.

Confusion Matrix

pred.Arbol	Neutro	Otro problema	Portabilidad	Positivo	Problema facturación	Problema técnico
Neutro	79	4	1	5	2	5
Otro problema	83	90	24	36	32	62
Portabilidad	3	1	15	1	1	0
Positivo	7	2	1	12	2	0
Problema facturación	0	1	2	0	11	0
Problema técnico	4	19	13	4	13	90
Total	176	117	56	58	61	157

Cuadro 3.4: Matriz de confusión Árbol decisión única variable

Overall Statistics

Accuracy: 0.4751
95% CI: (0.4353, 0.5152)
No Information Rate: 0.2825
P-Value [Acc > NIR]: < 2.2e-16

Kappa: 0.337
Mcnemar's Test P-Value: < 2.2e-16

Statistics by Class:	Neutro	Otro problema	Portabilidad	Positivo	Problema facturación	Problema técnico
Sensitivity	0.4489	0.7692	0.27273	0.19298	0.18033	0.5732
Specificity	0.9620	0.5316	0.98944	0.97880	0.99644	0.8863
Pues Pred Value	0.8229	0.2752	0.71429	0.47826	0.84615	0.6294
Neg Pred Value	0.8159	0.9088	0.93355	0.92333	0.91803	0.8604
Prevalence	0.2825	0.1878	0.08828	0.09149	0.09791	0.2520
Detection Rate	0.1268	0.1445	0.02408	0.01766	0.01766	0.1445
Detection Prevalence	0.1541	0.5249	0.03371	0.03692	0.02087	0.2295
Balanced Accuracy	0.7054	0.6504	0.63108	0.58589	0.58838	0.7298

Cuadro 3.5: Indicadores Árbol de decisión única variable

Podemos observar que lo resultados obtenidos son muy inferiores a los esperados, se obtuvo un accuracy de 0,4751. Se observa que la clase minoritaria presentan valores muy bajos para la sensitivity, los mismos fueron: portabilidad: 0,27273, positivos :0,19298 y problemas de facturación: 0,18033. Lo que nos dice que el modelo nos está realizando una mala predicción de VP.

3.2.1.2 SVM

Considerando los bajos valores arrojados en el modelo anterior se procederá a utilizar un segundo algoritmo, el mismo será SVM, donde utilizaremos las combinaciones posibles de Kernel, costo y Gamma se nos quedaremos con aquella que maximiza el accuracy. Los resultados posibles fueron las siguientes:

Se procederá a ver cuál es el costo y gamma que maximizan el accuracy:

	Kernel= Lineal	**Kernel= Lineal**	**Kernel= Lineal**	**Kernel= Lineal**	**Kernel= Polinomial**
	Cost=0,1-Gamma=0,1	**Cost=0,5-Gamma=0,5**	**Cost=1-Gamma=1**	**Cost=2-Gamma=2**	**Cost=1-Gamma=1**
Accuracy	0.6656	0,6945	0.6801	0.6768	0.5584

Cuadro 3.6: Costo y gamma que maximizan el accuracy

Comparando y analizando los primeros resultados obtenidos con SVM vemos una mejora en los resultados con solo cambiar el Kernel de polimonial a lineal, se observa

como con una Cost= 0,5 y Gamma=0,5 se maximiza en el valor del Accuracy. Obteniendo un valor de 0,6945.

Los resultados obtenidos con estos parámetros serán los siguientes:

pred.Arbol	Neutro	Otro problema	Portabilidad	Positivo	Problema facturación	Problema técnico
Neutro	149	27	2	19	4	8
Otro problema	11	60	5	8	5	18
Portabilidad	0	3	46	0	0	1
Positivo	7	6	1	26	4	6
Problema facturación	1	4	0	2	34	7
Problema técnico	8	17	1	2	13	117
Total	176	117	55	57	60	157

Cuadro 3.7: Matriz de confusión SVM única variable

Overall Statistics

Accuracy : 0.6945
95% CI : (0.6567, 0.7305)
No Information Rate : 0.283
P-Value [Acc > NIR]: < 2.2e-16

Kappa: 0.6116
Mcnemar's Test P-Valué: NA

	Neutro	Otro problema	Portabilidad	Positivo	Problema facturación	Problema técnico
Sensitivity	0.8466	0.51282	0.83636	0.45614	0.56667	0.7452
Specificity	0.8655	0.90693	0.99295	0.95752	0.97509	0.9118
Pos Pred Value	0.7129	0.56075	0.92000	0.52000	0.70833	0.7405
Neg Pred Value	0.9346	0.88932	0.98427	0.94580	0.95470	0.9138
Prevalence	0.2830	0.18810	0.08842	0.09164	0.09646	0.2524
Detection Rate	0.2395	0.09646	0.07395	0.04180	0.05466	0.1881
Detection Prevalence	0.3360	0.17203	0.08039	0.08039	0.07717	0.2540
Balanced Accuracy	0.8560	0.70988	0.91465	0.70683	0.77088	0.8285

Cuadro 3.8: Indicadores SVM única variable

Utilizando el algoritmo de SVM observamos que los valores de todos los indicadores se incrementaron de manera de considerable, obteniendo un accurracy de 0,6945 y los valores obtenidos en la sensibilidad aumentaron para todas las clases, pero observamos que las clases minoritarias siguen siendo bajos.

3.2.2 Modelo multi variable

A continuación, se procederá a separar la predicción en 2 etapas, en la primera corresponde a una primera variable con 3 clases, ellas serán positivos, neutros y negativos. Luego de realizada esta predicción nos quedaremos solo con los Tuis negativos y se realiza una nueva predicción, donde la misma tendrá 4 clases: Portabilidad, problemas técnicos, problemas de facturación y otros problemas.

3.2.2.1 Arboles de decisión para clasificación

Los resultados obtenidos para la primera variable son los siguientes:

Confusion Matrix and Statistics

Prediction	Negativo	Neutro	Positivo
Negativo	381	94	48
Neutro	10	82	4
Positivo	0	0	5
Total	391	176	57

Cuadro 3.9: Matriz de confusión Árbol de decisión Variable I

Overall Statistics

Accuracy: 0.7484
95% CI: (0.7124, 0.782)
No Information Rate: 0.6266
P-Value [Acc > NIR]: 6.509e-11

Kappa: 0.4147

Statistics by Class:

	Negativo	Neutro	Positivo
Sensitivity	0.9744	0.4659	0.07018
Specificity	0.3863	0.9688	1
Pos Pred Value	0.7271	0.8542	1
Neg Pred Value	0.9000	0.8220	0.91452
Prevalence	0.6266	0.2821	0.09135
Detection Rate	0.6106	0.1314	0.00641
Detection Prevalence	0.8397	0.1538	0.00641
Balanced Accuracy	0.6803	0.7173	0.53509

Cuadro 3.10: Indicadores árbol de decisión variable I

Se puede observar que los resultados para esta variable no son buenos, el accuracy presenta un valor de 0,7484. Este valor de precisión se presenta como aceptable pero el resto de los indicadores como sensitivity para la clase Neutro como positivo presenta valores inferiores, los que nos dice que el modelo nos está realizando una mala predicción de VP.

Resultado variable II

Una vez obtenidos los resultados de la primera variable se procederá a clasificar solo los Tuits Negativos con la apertura indicada para la segunda variable.

Para la segunda variable los resultados obtenidos serian:

Confusion Matrix and Statistics

Predicción	Otro problema	Portabilidad	Problema facturación	Problema técnico
Otro problema	99	14	23	48
Portabilidad	3	29	1	1
Problema facturación	1	2	18	3
Problema técnico	14	12	18	105
Total	117	57	60	157

Cuadro 3.11: Matriz de confusión Árbol de decisión Variable II

Overall Statistics

Accuracy: 0.6427
95% CI: (0.5928, 0.6903)
No Information Rate: 0.4036
P-Value [Acc > NIR]: < 2.2e-16

Kappa: 0.4761
Mcnemar's Test P-Value: NA

Statistics by Class:

	Otro problema	Portabilidad	Problema facturación	Problema técnico
Sensitivity	0.8462	0.50909	0.30000	0.6688
Specificity	0.6875	0.98503	0.98480	0.8103
Pos Pred Value	0.5380	0.84848	0.78261	0.7047
Neg Pred Value	0.9122	0.92416	0.88525	0.7833
Prevalence	0.3008	0.14139	0.15424	0.4036
Detection Rate	0.2545	0.07198	0.04627	0.2699
Detection Prevalence	0.4730	0.08483	0.05913	0.3830
Balanced Accuracy	0.7668	0.74706	0.64240	0.7396

Cuadro 3.12: Indicadores Árbol de decisión Variable II

Como se puede observar los resultados obtenidos con el modelo de predicción árbol de clasificación son bajos, arrojando para la primera variable un accuracy de 0.7484 y para la segunda variable 0.6427. Si analizamos los valores arrojados para sensibilidad y especificidad se observa valores muy bajos para por ejemplo la clase Positivos y neutros de la primera clasificación. Por lo dicho anteriormente se procederá a ver otros modelos que mejoren los valores ya encontrados y posteriormente a comparar los resultados obtenidos.

3.2.2.2 SVM

Resultado variable I

Considerando los bajos valores arrojados en el modelo anterior se procederá a utilizar un segundo algoritmo, el mismo será SVM, donde utilizaremos las combinaciones posibles de Kernel, costo y Gamma. Los resultados posibles fueron las siguientes:

Se procederá a ver cuál es el costo y gamma que maximizan el accuracy:

	Kernel= Lineal	**Kernel= Lineal**	**Kernel= Lineal**	**Kernel= Lineal**	**Kernel= Polinomial**
	Cost=0,1-Gamma=0,1	**Cost=0,5-Gamma=0,5**	**Cost=1-Gamma=1**	**Cost=2-Gamma=2**	
Accuracy	0.7788	0.7869	0.7821	0.7692	0.6731

Cuadro 3.13: Costo y gamma que maximizan el accuracy

Comparando y analizando los primeros resultados obtenidos con SVM vemos una mejora en los resultados con solo cambiar el Kernel de polimonial a lineal, se observa como con una Cost= 0.5 y Gamma=0.5 se maximiza en el valor del Accuracy

Matriz de confusión y demás valores que arrojan estos parámetros serán:

Confusion Matrix and Statistics

	Reference		
Prediction	**Negativo**	**Neutro**	**Positivo**
Negativo	335	27	22
Neutro	48	139	18
Positivo	8	10	17
Total	391	176	57

Cuadro 3.14: Matriz de confusión SVM Variable I

Overall Statistics

Accuracy: 0.7869
95% CI: (0.7526, 0.8184)
No Information Rate: 0.6266
P-Value [Acc > NIR]: < 2.2e-16

Kappa: 0.5874
Mcnemar's Test P-Value: 0.002093

Statistics by Class:

	Negativo	**Neutro**	**Positivo**
Sensitivity	0.8568	0.7898	0.29825
Specificity	0.7897	0.8527	0.96825
Pos Pred Value	0.8724	0.6780	0.48571
Neg Pred Value	0.7667	0.9117	0.93209
Prevalence	0.6266	0.2821	0.09135
Detection Rate	0.5369	0.2228	0.02724
Detection Prevalence	0.6154	0.3285	0.05609
Balanced Accuracy	0.8232	0.8212	0.63325

Cuadro 3.15: Indicadores SVM Variable I

Más allá que el valor del Accuracy es aceptable se observa que la sensibilidad para la clase positiva en muy baja (0.29825) lo que nos dice que no es buena la predicción de los de los VP, para el resto de las clases los valores son aceptables 0.7898 para neutros

y 0.8568 para negativos. La especificidad los valores son aceptables para las 3 clases o sea que nos estima bien nuestro FN.

Se aplicará el balanceo de clase (Oversampling) que consiste en modificar la distribución de los datos incrementando el número de la clase minoritaria, esta técnica solo será aplicada para el conjunto de Train. Luego se procederá a analizar los resultados para ver si mejoran los valores de sensibilidad de la clase Positiva. Para nuestro caso la clase minoritaria seria los positivos donde antes de aplicar el balanceo de clase tenía 57 casos y luego del balanceo quedaron casos, para la aplicación de esta técnica se utilizará SMOTE, tal como lo indica en su trabajo realizado por Arnejo Calviño [3]: SMOTE (Synthetic Minority Oversampling Method): Esta técnica genera nuevas instancias de la clase minoritaria interpolando los valores de las instancias minoritarias más cercanas a una dada.

Antes de aplicar el Oversampling la distribución del set de datos train (1876 tuits que corresponde al 75% de los 2.500 Tuits) era de la siguiente manera:

Negativo	Neutro	Positivo
1173	530	173

Cuadro 3.16: Distribución de Tuits antes de Oversampling

Luego de aplicar el balanceo de clase quedaría la siguiente distribución:

Negativo	Neutro	Positivo
1292	583	432

Cuadro 3.17: Distribución de Tuits después de Oversampling

A continuación, se expondrán los resultados obtenidos aplicando la técnica oversampling para set de datos train.

Confusion Matrix and Statistics

Prediction	Negativo	Neutro	Positivo
Negativo	338	27	18
Neutro	39	137	17
Positivo	14	12	22
Total	391	176	57

Cuadro 3.18: Matriz de confusión variable I con Oversampling

overall Statistics

Accuracy : 0.7965
95% CI : (0.7627, 0.8274)
No Information Rate : 0.6266
P-Value [Acc > NIR] : <2e-16

Kappa : 0.6095
Mcnemar's Test P-Value : 0.3151

Statistics by Class:

	Negativo	Neutro	Positivo
Sensitivity	0.8645	0.7784	0.38596
Specificity	0.8069	0.8750	0.95414
Pos Pred Value	0.8825	0.7098	0.45833
Neg Pred Value	0.7801	0.9095	0.93924
Prevalence	0.6266	0.2821	0.09135
Detection Rate	0.5417	0.2196	0.03526
Detection Prevalence	0.6138	0.3093	0.07692
Balanced Accuracy	0.8357	0.8267	0.67005

Cuadro 3.19: Indicadores SVM Variable I con Oversampling

Observamos que con el balanceo de las clases hubo una mejora en el accuracy, pasó de 0.7869 a 0.7965 y la sensitivity de la clase positiva mejoro, paso de 0.29825 a 0.38596.

Optimización de hiperparámetros- cross-validation

El método de *SVM* lineal tiene un único hiperparámetro *C* que establece la penalización por clasificación incorrecta, regulando así el balance entre bias y varianza. Al tratarse de un hiperparámetro, su valor óptimo no se aprende en el proceso de entrenamiento, para estimarlo hay que recurrir a validación cruzada.

A diferencia de la técnica de entrenamiento utilizada en puntos anteriores, o sea en la división entre Test y Train para aplicar Cross validation utilizaron la totalidad del Data set.

Se usarán 10 Folds, los resultados obtenidos son los siguientes:

```
Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  0.5
      gamma:  0.5

Number of Support Vectors:  1665

 ( 760 526 379 )


Number of Classes:  3

Levels:
 Negativo Neutro Positivo

10-fold cross-validation on training data:

Total Accuracy: 87.44715
Single Accuracies:
 89.25081 87.66234 85.66775 87.98701 87.62215 85.06494 87.94788 88.311
69 88.27362 86.68831
```

La Validación Cruzada vamos a hacer uso de la función createFolds del paquete caret, y luego entrenaremos cada modelo sobre k= 10 subconjuntos, donde el valor total del Accuracy es de 87.44 y a modo de ejemplo para el primer Fold en valor de accuracy es de 89.25. La utilización de esta técnica es solo con los fines de la validación de los

resultados ya obtenidos ya que se aplicó oversampling a todo el data set que provoco la mejora en los resultados.

Resultado variable II

Una vez realizada la primera clasificación, que nos permitió obtener los Tuits negativos, se procederá a la apertura de los mismos, la misma contará con 4 clases: Portabilidad, mal servicio, problemas de facturación, otros negativos y se descomponer de la siguiente manera:

La distribución de los 1.564 Twitter será:

Otro problema	Portabilidad	Problema facturación	Problema técnico
468	222	243	631

Cuadro 3.20: Distribución de Tuis Negativos

Se procederá a ver cuál es el costo y gamma que maximizan el accuracy:

	Kernel= Lineal Cost=0,1- Gamma=0,1	Kernel= Lineal Cost=0,5- Gamma=0,5	Kernel= Lineal Cost=1- Gamma=1	Kernel= Lineal Cost=2- Gamma=2	Kernel= Polinomial
Accuracy	0.7252	0.7275	0.7224	0.7301	0.6093

Cuadro 3.21: Costo y gamma que maximizan el accuracy

Comparando y analizando los primeros resultados obtenidos con SVM vemos una mejora en los resultados con solo cambiar el Kernel de polimonial a lineal, se observa como con una Cost= 2 y Gamma=2 se maximiza en el valor del Accuracy

Matriz de confusión y demás valores que arrojan estos parámetros serán:

Confusion Matrix and Statistics

	Otro_problema	Portabilidad	Problema facturación	Problema técnico
Otro problema	88	7	8	28
Portabilidad	3	44	3	2
Problema facturación	3	2	36	11
Problema técnico	23	4	13	116
Total	117	57	60	157

Cuadro 3.22: Matriz de confusión SVM Variable II

Overall Statistics

Accuracy : 0.7301
95% CI : (0.683, 0.7736)
No Information Rate : 0.4036
P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.6134

Statistics by Class:	Otro_problema	Portabilidad	Problema_facturacion	Problema_tecnico
Sensitivity	0.7521	0.8000	0.60000	0.7389
Specificity	0.8419	0.9760	0.95745	0.8276
Pos Pred Value	0.6718	0.8462	0.72000	0.7436
Neg Pred Value	0.8876	0.9674	0.92920	0.8240
Prevalence	0.3008	0.1414	0.15424	0.4036
Detection Rate	0.2262	0.1131	0.09254	0.2982
Detection Prevalence	0.3368	0.1337	0.12853	0.4010
Balanced Accuracy	0.7970	0.8880	0.77872	0.7832

Cuadro 3.23: Indicadores SVM Variable II

Se aplicará el balanceo de clase Oversampling que consiste en modificar la distribución de los datos incrementando el número de la clase minoritaria. Luego se procederá a analizar los resultados para ver si mejoran los valores de Sensitivity.

Antes de aplicar el Oversampling la distribución del set de datos train era de la siguiente manera:

Otro problema	Portabilidad	Problema facturación	Problema técnico
351	167	183	474

Cuadro 3.24: Distribución de Tuits antes de Oversampling

Luego de aplicar el balanceo de clase quedaría la siguiente distribución:

Otro problema	Portabilidad	Problema facturación	Problema técnico
400	500	212	535

Cuadro 3.25: Distribución de Tuits con Oversampling

Los resultados que arroja luego de aplicar el Oversampling serían los siguientes:

Prediction	Otro problema	Portabilidad	Problema facturación	Problema técnico
Otro problema	93	0	12	19
Portabilidad	4	55	1	3
Problema facturación	3	0	37	8
Problema técnico	17	0	10	127
Total	117	55	60	157

Cuadro 3.26: Matriz de confusión SVM Variable II con Oversampling

```
verall Statistics

               Accuracy : 0.8021
                 95% CI : (0.7589, 0.8405)
    No Information Rate : 0.4036
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7191
```

	Otro_problema	Portabilidad	Problema facturación	Problema técnico
Sensitivity	0.7949	1	0.61667	0.8089
Specificity	0.8860	0.9940	0.96657	0.8836
Pos Pred Value	0.7500	0.9649	0.77083	0.8247
Neg Pred Value	0.9094	1	0.93255	0.8723
Prevalence	0.3008	0.1414	0.15424	0.4036
Detection Rate	0.2391	0.1414	0.09512	0.3265
Detection Prevalence	0.3188	0.1465	0.12339	0.3959
Balanced Accuracy	0.8405	0.9970	0.79162	0.8463

Cuadro 3.27: Indicadores SVM Variable II con Oversampling

Cross Validation: Se usarán 10 Folds, los resultados obtenidos son los siguientes:
Parameters:
SVM-Type: C-classification
SVM-Kernel: linear
cost: 2
gamma: 2

Number of Support Vectors: 1308

(419 424 228 237)

Number of Classes: 4

Levels:
Neutro Otro_problema Portabilidad Positivo Problema_facturacion Problema_tecnico

10-fold cross-validation on training data:

Total Accuracy: 83.07164
Single Accuracies:
81.7734 81.37255 81.37255 84.80392 79.41176 89.65517 85.78431 82.84314 82.352
94 81.37255

La Validación Cruzada vamos a hacer uso de la función createFolds del paquete caret, y luego entrenaremos cada modelo sobre k= 10 subconjuntos, donde el valor total del Accuracy es de 83.07 y a modo de ejemplo para el primer Fold en valor de accuracy es de 81.77.

Las métricas obtenidas con el algoritmo de SVM son muy superiores a las resultantes al modelo árbol de decisión, a modo de ejemplo para la primera variable, el Accuracy en árbol es de 0,7484, en cambio para SVM utilizando las mismas variables en resultado es de 0,7965. Para la segunda variable se obtuvo un Accuracy de 0,6427, ese resultado es inferior al obtenido en SVM que fue de 0,8021. Por lo dicho anteriormente se usara el Algoritmo SVM ya que el mismo optimiza los paramentos de predicción.

4.1 Conclusión y trabajos futuros

El presente trabajo permite predecir los comentarios de los usuarios más propensos a realizar la portabilidad numérica en empresas de telefonía móvil. Para ello se utilizó como fuente de datos los Tuits de Movistar Argentina y con ellos se creó un modelo de predicción de SVM, con lo cual se obtuvo un accuracy del 0.7965 para la primera variable y del 0.8021 para la segunda variable. En el mismo podremos identificar cual es la disconformidad de aquellos usuarios que tiene intención de cambiar de compañía. Esta información nos permitirá segmentar las campañas de retención de los clientes y poder cubrir las necesidades del mismo sin que lleguen a tomar esa decisión.

En el presente trabajo se comprobó que la utilización de las variables por separadas mejoro los resultados obtenidos como así también técnicas para desbalanceo de clase como Oversampling también contribuyen a la optimización de resultados. También se demostró que el algoritmo SVM funciono mejor para la predicción de Tuits comparados con Arboles de decisión para clasificación, también se decidió la no utilización de Redes neuronales ya el tiempo de procesamiento era muy superior a SVM y los resultados obtenidos eran inferiores.

En trabajos futuros se procederá a realizar distintos clúster para los distintos tipos de comentarios y los podremos agrupar de acuerdo a su tipo. De esta forma podremos entender más la problemática de cada uno de los clientes.

5.1 Referencias – Bibliografía

[1] Ley Resolución 67/2011 (Boletín Oficial Nº 32.192, 15/07/11) Régimen de Portabilidad Numérica. Modificase la Resolución Nº 98/10. Publishing Bs. As., 14/6/2011

[2] Ente nacional de Comunicaciones [ENACOM] Ministerio de Modernización. Biblioteca Especializada Web.

https://www.enacom.gob.ar/informes-de-mercado_p2877

http://datosabiertos.enacom.gob.ar/visualizations/29884/portabilidad-numerica-movil-portaciones-netas-mensuales-por-operador/

http://datosabiertos.enacom.gob.ar/dataviews/241761/portabilidad-numerica-movil-altas-mensuales-por-operador/

[3] Hugo Antonio Arnejo Calviño (2017) Métodos para la mejora de predicciones en clases desbalanceadas en el estudio de bajas de clientes (CHURN). Master en Técnicas Estadísticas USC-Universidad da Coruña.

[4] Patricio Alfredo Pérez Villanueva (2014) Modelo de predicción de Fuga de Cliente de telefonía Móvil Post pago, Memoria para optar al Tıtulo de Ingeniero Civil Industrial. Departamento de Ingeniería Industrial, Universidad de Chile, Chile.

[5] A. Melgarejo Galván, K., Clavo Navarro, R. (2017) Big Data Architecture for Predicting Churn Risk in Mobile Phone Compañías. Publisher: Springer International Publishing 2017.p 106-116

[6] Eric Chiang (2014) Predicting customer churn with scikit-learn. The Yhat Blog

http://blog.yhat.com/posts/predicting-customer-churn-with-sklearn.html

[7] Francisco Parra (2017) Estadística y Machine Learning con R. RPubs Blog

[8] Joaquín Amat Rodrigo (2017) Support Vector Machines, SVMs. Publisher: RPubs Blog

[9] Matthew A. Russell (2013) Mining the Social Web. Second Edition. Published by O'Reilly Media, Inc, 1005 Gravenstein Highway North, Sebastopol.

[10] Francisco Barrientos y Sebastián A. Ríos (2013) Aplicación de Minería de Datos para predecir Fuga de Clientes en la Industria de las Telecomunicaciones, Revista Ingeniería de Sistemas, vol. XXVII, p. 73-107.

[11] Ley Nº 25.326. Ley de Protección de los Datos Personales. Informacion legislativa. Ministerio de justicia y derechos humanos. Presidencia de la nación.

[12] Overview .Tweet updates (2018) Twitter, Inc.

[13] Ali R., Omar S. Al-Kadi, Hossam F. (2014) a Support Vector Machine Approach for Churn Prediction in Telecom Industry. Publisher: ResearchGate p.3962-3971

[14] Georgios Paltoglou M. T. (2012) Twitter, MySpace, Digg: Unsupervised Sentiment Analysis in Social Media ACM Trans. Intell. Syst. Technol . Published 2012 in ACM TIST. SemanticScholar

[15] John W. Foreman (2014) DATA SMART. Published by Jhon Wiley & Sons,Inc.

[16] Amit Agarwal (2015), How to Save Tweets for any Twitter Hashtag in a Google Sheet. Blog: Digital inspiration

https://www.labnol.org/internet/save-twitter-hashtag-tweets/6505/

[17] https://es.wikipedia.org/wiki/Portabilidad_num%C3%A9rica

"Las páginas web de este trabajo fueron visitadas en el mes de diciembre de 2018"