Arquitectura de Big Data para la Predicción de la Portabilidad Numérica en Empresas de Telefonía Móvil

Alonso Raúl Melgarejo Galván
Facultad de Ingeniería de Sistemas
Universidad Nacional Mayor de San Marcos
Lima - Perú
alonsoraulmg@gmail.com

Katerine Rocio Clavo Navarro
Facultad de Ingeniería de Sistemas
Universidad Nacional Mayor de San Marcos
Lima - Perú
perclavo@gmail.com

Abstract

Actualmente en el Perú, las compañías de telefonía móvil se han visto afectadas por el problema de la portabilidad numérica, puesto que, desde julio del 2014, los clientes pueden cambiar de operadora móvil en sólo 24 horas. Las compañías buscan soluciones analizando los datos históricos de sus clientes para generar modelos de predicción e identificar qué clientes portarán, sin embargo, la actual forma en la que se realiza esta predicción es demasiado lenta. En el presente trabajo, mostramos una arquitectura de Big Data que resuelve los problemas de las arquitecturas "clásicas" y aprovecha los datos de las redes sociales para predecir en tiempo real qué clientes son propensos a portar a la competencia según sus opiniones. El procesamiento de los datos es realizado por Hadoop el cual implementa MapReduce y permite procesar grandes cantidades de datos en forma paralela. La arquitectura también utiliza otras herramientas de Hadoop como Mahout para generar nuestro modelo de predicción, y Hive para gestionar los datos con una sintaxis similar a SQL. Al realizar las pruebas y observar los resultados, logramos obtener un alto porcentaje de precisión (90.03% de aciertos), y procesar 10'000 comentarios en 14 segundos.

1 Introducción

En el Perú, la pérdida de clientes en la industria de la telefonía móvil es un problema que actualmente afecta a las grandes empresas de telecomunicaciones del país debido a la fuerte competencia que se ha generado en el mercado de servicios móviles de voz y datos, y que ha generado grandes ofertas comerciales y guerra de precios. OSIPTEL (2015) menciona que desde el ingreso de las operadoras móviles Bitel y Entel, la competencia se ha incrementado, siendo Entel la operadora que más clientes ha obtenido de Claro Perú y de Movistar.

Como vemos en la Figura 1, de acuerdo a las últimas cifras de OSIPTEL (2015), se muestra que en marzo, la portabilidad móvil creció en 46%, alcanzando un récord de 65,142 portaciones, el más alto desde julio del año pasado, fecha en la que se relanzó el mecanismo para realizar una portabilidad numérica y cambiar de operadora móvil en solo 24 horas manteniendo el mismo número de teléfono.

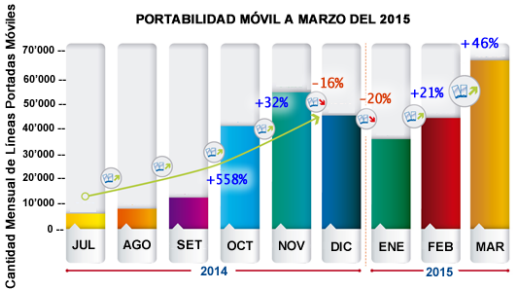

Figura 1: Portabilidad móvil a marzo del 2015

Como (Francisco et al., 2013) menciona, en el área de las telecomunicaciones, la fuga de clientes es un problema que cada vez es más necesario estudiar debido a la alta competitividad que se está desarrollando a nivel mundial. Se hace necesaria la aplicación de herramientas avanzadas que permitan predecir y describir de algún modo, qué clientes tienen mayor potencial de riesgo de cambiarse de compañía.

2 El problema de la portabilidad numérica: La fuga de clientes

La pérdida de clientes es un riesgo costoso que si no se maneja con cuidado podría llevar a la

compañía móvil de rodillas.

Según define (Francisco et al., 2013), dentro del sector de telecomunicaciones, el término churn o fuga de clientes es usado para describir el cese de servicios de la suscripción de un cliente, y se habla de churner o fugado para denominar a un cliente que ha dejado la compañía. Como (Clement et al., 2013) menciona, un cliente puede renunciar a la empresa e iniciar la terminación de su contrato de servicios (churn voluntario), o bien la empresa puede expulsarlo por fraude, falta de pago o por subutilización de los servicios suscritos (churn involuntario).

(Clement et al., 2013) también menciona que la fuga de clientes puede llegar a ser muy costosa para una compañía, ya que el cliente fuga hacia la competencia y por ende, no solo se pierde el ingreso no percibido, sino también el prestigio de la compañía expresado en la participación de mercado de la competencia.

2.1 Causas que ocasionan la fuga de clientes

Los factores que contribuyen al comportamiento de fuga en los servicios de telecomunicaciones son diversos.

(Francisco et al., 2013) menciona que, dentro de las principales razones por las que un cliente deja de adquirir los productos de una compañía se encuentran: la disconformidad con la calidad de la señal, cobertura, servicio al cliente, precios, cobros irregulares y la falta de políticas de retención con un mejor trato a los clientes, pero por otro lado Patricio (2014) también menciona que la red de contactos donde se desenvuelve el cliente es muy importante, pues el efecto "boca a boca" se ha convertido en un factor determinante en las decisiones de compra de un consumidor.

Anticiparse a esta problemática y lograr identificar qué lleva a un cliente a terminar su contrato, entrega diversos beneficios a la compañía como por ejemplo, la menor inversión en retener a un cliente (gracias a la recomendación de la red de contactos). Patricio (2014) afirma que adquirir un nuevo cliente cuesta seis veces más que retener a uno propio, y que los clientes que se mantienen más tiempo en la empresa generan mayores ingresos y son menos sensibles a las acciones de marketing de la competencia convirtiéndose en consumidores menos costosos de servir.

2.2 La influencia de la opinión dejada en la web

Hoy en día, los avances tecnológicos como el internet y las redes sociales permiten que el cliente tenga mayor acceso a la información y pueda comparar fácilmente la compañía que más le convenga.

Bajo el contexto de la competitividad que actualmente se vive en el mercado de telecomunicaciones, y sumándose a ello, el mayor acceso a la información, Patricio (2014) afirma que se genera un marco de flexibilidad y dinamismo respecto a la movilidad de los clientes de una compañía a otra.

Como menciona Tania (2011), la dinámica de participación social y la influencia que pueden tener las valoraciones dejadas por los consumidores en internet, han hecho que las empresas del mercado presten atención a la gestión de las opiniones que se dejan en la web sobre ellas. El community manager de una empresa debe ser rápido en la resolución de los conflictos que percibe en la web. Un conflicto que tarda un día en resolverse, probablemente se convertirá en un conflicto no resuelto, y en muchos casos propiciará una fuga de clientes hacia la competencia, afectando igualmente la reputación online de la empresa. El community manager debe tener criterio para destacar aquellos comentarios positivos, negativos o notables, que por alguna razón, merezcan la ejecución de alguna estrategia especial.

Patricio (2014) menciona que las redes sociales pueden influenciar distintos aspectos de una persona como por ejemplo contratar un servicio, comprar un producto o abandonar una compañía, mediante el efecto "boca a boca".

(Yiou et al., 2015) también afirma que las opiniones de los clientes tienen un impacto en los productos y los servicios, por eso es necesario capturar estas opiniones por medio de calls centers, correos, cuestionarios o webs para entender las necesidades de los clientes. Es por esto que el explotar la "voz del consumidor" debe ser considerado en la predicción de abandono de clientes.

2.3 La solución actual

Actualmente las empresas de telefonía móvil llegan a desarrollar modelos predictivos para identificar a los clientes que pueden llegar a portar hacia la competencia. La construcción de estos modelos puede variar, pero en general, se siguen los pasos mostrados en la Figura 2, como nos explica

(Clement et al., 2013) a continuación.

Primero se deben identificar las fuentes de datos con las que se construirá el modelo de predicción, la cuales corresponden a los datos internos de la empresa referentes al perfil del cliente y el tráfico de llamadas.

El perfil del cliente describe el grupo demográfico de los clientes y las características que tienen en común respecto al segmento, el plan contratado, el tipo de pago, el riesgo crediticio, el área demográfica y las penalidades por no pagar las cuentas. El tráfico de llamadas describe el tráfico generado y recibido por un cliente, el cual es caracterizado por las llamadas realizadas y recibidas desde líneas fijas o móviles, locales o internacionales, desde qué operador se realizó, la cantidad de SMS enviados y recibidos, y el tráfico de red en internet generado.

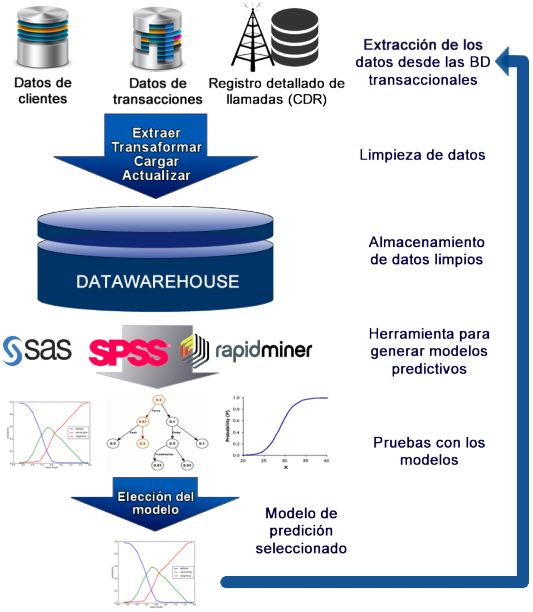

Figura 2: Pasos en el desarrollo de modelos de predicción

Luego de identificar las fuentes de datos con las que se trabajará, debe realizarse una extracción y limpieza de ellos. La limpieza de datos consiste en encontrar errores en los datos, duplicidad, incoherencias o valores incompletos e inconsistentes. Esta información es corregida reemplazando valores generados, o eliminada en el peor de los casos. Todos estos datos son almacenados en un Datawarehouse.

Una vez que los datos han sido limpiados y están listos para ser usados, se utilizan diferentes algoritmos de minería de datos existentes como Naive Bayes, árboles de decisión o redes bayesianas, para construir modelos de predicción. Cuando los modelos son generados, se realizan pruebas sobre ellos para encontrar aquel que tenga el mejor ratio de predicción, el cual será el modelo a usar para predecir la portabilidad.

Por supuesto, este no es el paso final, ya que el comportamiento de los clientes puede cambiar a lo largo del tiempo, por lo que es necesario actualizar el modelo volviendo a repetir todos los pasos.

Ahora que entendemos cómo se generan actualmente los modelos de predicción, veamos cuáles son los problemas que poseen.

2.4 Problemas de la solución actual

Dado el entorno actual en el cual las empresas de telefonía móvil se desenvuelven, la solución que actualmente utilizan para predecir la fuga de clientes presenta cuatro problemas:

P1. Portabilidad rápida, predicción lenta: Como menciona OSIPTEL (2015), la nueva ley de portabilidad hace que los clientes puedan realizar una portabilidad hacía la competencia en 24 horas, así que se necesita predecir esta portabilidad de manera rápida para evitarlo. La solución clásica requiere mantener un Datawarehouse con datos limpios y filtrados, lo cual consume demasiado tiempo.

P2. Confidencialidad de datos: La solución clásica trabaja principalmente con los datos internos de la empresa puesto que están estructurados y disponibles, sin embargo, debido a la confidencialidad y privacidad del negocio de las telecomunicaciones, es muy difícil para las operadoras encontrar fuentes de datos públicas que sean fidedignas y puedan usarse como un input adicional en la predicción de la portabilidad de sus clientes, como menciona (Clement et al., 2013).

P3. Datos no estandarizados e incoherentes: Estandarizar las características de las diversas fuentes de datos usadas para realizar un análisis es un reto ya que consume demasiado tiempo y esfuerzo. Es necesario implementar y mantener un Datawarehouse para realizar análisis de datos, lo cual incluye eliminar información irrelevante, duplicada o con valores nulos como menciona (Clement et al., 2013).

P4. Opinión cambiante: No se analiza la opinión cambiante que tienen los clientes respecto a un servicio, lo cual es un factor determinante

en sus decisiones como menciona Patricio (2014). Para poder analizar el cambio en las opiniones, deben incluirse fuentes de datos adicionales a los perfiles de clientes y el tráfico de llamadas.

Estos cuatro problemas ocasionan que la identificación de clientes portadores sea lenta e inexacta, pues se requiere realizar una limpieza de los datos o un análisis y ordenamiento de sus fuentes, lo cual demuestra que la implementación de una arquitectura tradicional para predecir la portabilidad numérica tiene carencias que deben resolverse con un enfoque diferente.

3 Un enfoque diferente: Big Data

La necesidad de procesar grandes volúmenes de datos en tiempo real es crucial para las empresas de hoy en día, por ejemplo, el procesamiento de volúmenes masivos de datos donde se esconde información valiosa respecto al comportamiento de compra de productos o servicios de clientes, y poder generar nuevos productos analizando dichos comportamientos. Esto último es particularmente cierto en el mercado de los negocios de las telecomunicaciones, donde el número de clientes normalmente llega a varios millones como menciona (Francisco et al., 2013).

En estos escenarios es donde entra Big Data, el cual es un campo emergente de tratamiento de datos que permite analizar grandes cantidades de datos y maximizar el valor del negocio dando un soporte en tiempo real a la información necesaria para la toma de decisiones, según define (M. Vasuki et al., 2014). Por otro lado, (Kankana et al., 2014) también nos dice que la invención de nuevas tecnologías ha llevado a la utilización de grandes cantidades de datos que van en aumento, además se ha creado la necesidad de conservar, procesar y extraer estos datos. De esto se encarga el término Big Data.

Para tener un mayor conocimiento sobre Big Data, en esta sección se verá el concepto de las 5 V de Big Data, el aprovechamiento de las nuevas fuentes de datos, el procesamiento paralelo masivo que ofrece MapReduce y la implementación más popular que tiene llamada Hadoop.

3.1 Las 5 V

Big Data se caracteriza tradicionalmente por el concepto de las 3 V: volumen, variedad y velocidad como menciona (Mario et al., 2013), pero (Abdullah et al., 2015) también menciona que actualmente, se adicionan 2 V más: variabilidad y veracidad, dando así un total de 5 V.

El **volumen**, según (Mario et al., 2013), es la dimensión más obvia al caracterizar grandes colecciones de datos creadas para diferentes usos y propósitos. El almacenamiento de Big Data supone el reto más inmediato, ya que su primera responsabilidad es la de preservar todos los datos generados en el ámbito de los sistemas transaccionales. La decisión de cómo se almacenan los datos tiene un impacto considerable en el rendimiento de los procesos de recuperación, procesamiento y análisis de Big Data.

La **velocidad**, según (Mario et al., 2013), caracteriza los flujos de datos desarrollados en entornos cada vez más distribuidos. Se pueden distinguir dos tipos de flujos: los flujos de nuevos datos y los flujos que contienen resultados generados por consultas. La velocidad describe lo rápido que se generan, demandan y entregan los datos en su entorno de explotación.

La **variedad**, según (Mario et al., 2013), se refiere a los diferentes grados de estructura o falta de ella que pueden encontrarse en una colección de datos. La colección puede integrar datos procedentes de múltiples fuentes, por ejemplo: redes de sensores, logs generados en servidores, redes sociales, datos de origen político, económico o científico, entre otros. Cada una de estas fuentes de datos tiene esquemas diferentes que son difícilmente integrables en un modelo único, por lo tanto, el manejo efectivo de la variedad pasa por encontrar un modelo lógico que facilite la integración de los datos, independientemente de su estructura.

La **veracidad**, según (Abdullah et al., 2015), hace referencia a la incertidumbre que hay en las fuentes de datos y su nivel de confiabilidad. La veracidad de una fuente de datos disminuye si en ella existen inconsistencias y datos incompletos. Es necesario trabajar con datos precisos, no falsos, no corruptos y que provengan de una fuente de datos fidedigna.

Por último, la **variabilidad**, según (Abdullah et al., 2015) se refiere al cambio que tienen los datos a lo largo del tiempo, tanto en su estructura como en su contenido.

También hay que considerar lo que recomienda (Mario et al., 2013): cualquier arquitectura diseñada para la gestión de Big Data debe afrontar las variables anteriores, sin embargo, la decisión

de cuál de ellas afrontar en primer lugar depende del entorno de explotación final de la arquitectura, por ejemplo, optimizar el almacenamiento de datos es un aspecto más crítico para una arquitectura destinada a un dispositivo móvil, que para una que será ejecutada en un servidor de alto rendimiento; la velocidad con la que se recuperan los datos es una prioridad para una arquitectura en tiempo real, pero no lo es tanto para una de procesamiento en batch. Por lo tanto, una arquitectura para Big Data debe priorizar las cinco dimensiones anteriores con el objetivo de cubrir de forma efectiva los requisitos con los que se diseña.

3.2 Nuevas fuentes de datos: Web y Redes Sociales

Los datos en forma de texto dentro de internet están creciendo cada vez más y es imposible analizar estos datos de forma manual debido a su ingente cantidad. Es aquí donde la necesidad de la automatización se hace evidente. (Sunil et al., 2014) dice que en la web los usuarios tienen la oportunidad de expresar sus opiniones personales sobre tópicos específicos dentro de blogs, foros, sitios de revisión de productos y redes sociales.

Como indica (M. Vasuki et al., 2014), este crecimiento explosivo de la información textual en la web ha traído un cambio radical en la vida humana. En la web la gente comparte sus opiniones y sentimientos, lo cual crea una gran colección de opiniones y puntos de vista en forma de texto que pueden ser analizados para conocer la eficacia de los productos y los servicios.

3.3 Procesamiento paralelo: MapReduce

Es un framework de programación creado por Google para computación distribuida que utiliza el método de "Divide y Vencerás" para analizar grandes conjuntos de datos complejos y que garantiza la escalabilidad lineal. Utiliza dos funciones para procesar los datos: la función Map y la función Reduce. El funcionamiento de estas dos funciones puede verse en la Figura 3.

Como explica (G. Bramhaiah et al., 2015), la función Map divide los datos de entrada en subpartes más pequeñas. Estas partes son distribuidas a lo largo de los servidores para que sean procesadas por separado. Luego la función Reduce recolecta las respuestas de todas las subpartes y las combina en una única salida. MapReduce divide el procesamiento de un algoritmo en etapas paralelizables que se ejecutan en muchos nodos,

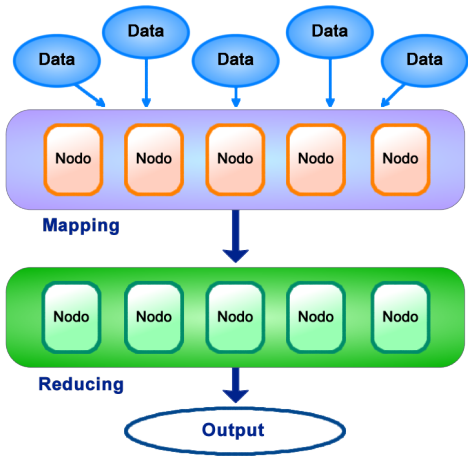

Figura 3: Procesamiento paralelo de datos con MapReduce

así como en etapas de agregación donde los datos obtenidos en la fase previa son procesados en un único nodo. De esta manera se facilita el almacenamiento y procesamiento del llamado Big Data. La herramienta más extendida según (Pablo et al., 2014) basada en el modelo MapReduce es Hadoop, la cual se explica a continuación.

3.4 La plataforma Hadoop

Según Dhruva (2014), Hadoop es un framework open-source para el almacenamiento y procesamiento de grandes cantidades de datos, sobre clústers que funcionen sobre hardware comodity.

(Debajyoti et al., 2014) agrega que una plataforma típica de Big Data basada en Hadoop incluye el sistema distribuido de archivos HDFS, el framework MapReduce de computación paralela, un sistema de alto nivel para la administración de datos como Pig o Hive, y Mahout como el módulo para el análisis de datos. Todo lo anterior es mostrado en la Figura 4.

(Kankana et al., 2014) explica que el HDFS es básicamente una estructura maestro/esclavo. Al maestro se le conoce como "Name Node", y a los esclavos como "Data Nodes". El trabajo principal del "Name Node" es almacenar metadatos de los datos, esto incluye la localización de los archivos que los contienen y también los diferentes atributos de los documentos. Los "Data nodes" se encargan de almacenar los datos en bloques en los diferentes nodos del clúster.

MapReduce, según (Debajyoti et al., 2014) es un módulo que Hadoop incorpora para el procesamiento paralelo de datos. Para que los progra-

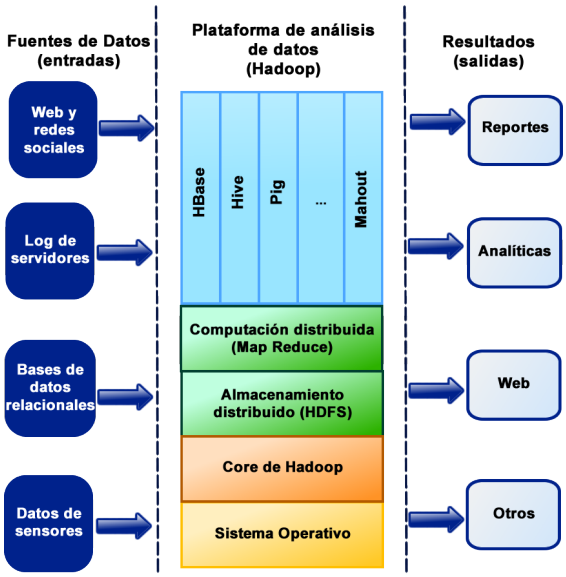

Figura 4: Plataforma Hadoop

madores puedan escribir programas sobre Hadoop deben de especificar las funciones Map y Reduce para que Hadoop las ejecute. Hadoop divide en muchos pequeños fragmentos la entrada la cual distribuye y procesa de forma paralela los nodos del clúster por medio de la función Map, luego por medio de Reduce combina los resultados. Cuando el procesamiento finaliza, el resultado puede residir en múltiples archivos.

Hive, según (Rakesh et al., 2014) es un módulo de Hadoop que soporta el manejo de archivos sobre HDFS por medio una sintaxis similar a SQL, el "Hive Query Language". Las consultas de Hive utilizan el módulo de MapReduce de Hadoop para ejecutarse de manera paralela y ocultar su complejidad al programador. Gracias a esto, es posible usar sencillas sentencias sobre los archivos ubicados en HDFS como "CREATE TABLE", "INSERT", "UPDATE", "SELECT", "DELETE", "JOIN", "GROUP BY", y otras sentencias válidas en el SQL clásico.

Mahout, según (Seyyed et al., 2014) ha sido diseñado para propósitos de minería de datos dentro de Hadoop, pues implementa los algoritmos de clustering y regresión más comunes por medio de MapReduce. Además Dhruva (2014) menciona que Mahout provee herramientas para encontrar de manera automática patrones en grandes volúmenes de datos y hace mucho más fácil y rápido el análisis dentro de Big Data.

4 El problema de la portabilidad numérica desde la perspectiva de Big Data

Después de haber explicado el problema de la portabilidad numérica y el enfoque de Big Data frente a las soluciones tradicionales, veremos cómo Big Data ofrece una solución para cada uno de los problemas encontrados en la portabilidad numérica. Analizaremos cómo cada problema está relacionado con una V de Big Data.

La Figura 5 muestra el cuadro en el que se relacionan a las 5 V de Big Data con cada uno de los problemas encontrados en la subsección 2.4. La explicación del por qué hacemos esta relación entre los problemas y las V es dada a continuación:

	Volumen	Velocidad	Variedad	Variabilidad	Veracidad
P1		X			
P2			X		X
P3	X		X		
P4				X	

Figura 5: Relación entre los problemas de la solución actual y las 5 V

El **primer problema** referido a la rapidez con la que un cliente puede fugar hoy en día a la competencia está relacionado a la V de "velocidad", puesto que se necesita identificar a estos clientes de inmediato para evitar que vayan a la competencia, y como solución, Big Data procesa y entrega la predicción en tiempo real.

El **segundo problema** referido a la confidencialidad de los datos está relacionado a las V de "variedad" y "veracidad" puesto que las soluciones clásicas trabajan principalmente con los datos internos de la empresa, ignorando las fuentes públicas como los sitios web que contienen datos no estructurados pero muy valiosos. Por supuesto que si la empresa decide utilizar datos públicos para realizar sus predicciones, debe tener la certeza de que éstos son fidedignos. Como solución, Big Data permite trabajar con nuevas fuentes de datos como las opiniones dejadas por los clientes en las redes sociales de la compañía móvil acerca del servicio, y poder predecir quienes tienen intención de portar.

El **tercer problema** referido a los datos no estandarizados e incoherentes, está relacionado a las V de "volumen" y "variedad". Los datos usados deben estar limpios y estandarizados antes de

comenzar un proceso de análisis; no puede trabajarse con los datos transaccionales directamente, sino que deben formatearse primero. Como solución, el enfoque de Big Data procesa los datos sin necesidad de realizar un proceso de limpieza.

Por último, el **cuarto problema** referido a la opinión cambiante de los clientes acerca del servicio vendría a estar relacionado con la V de "variabilidad", pues el enfoque de Big Data detecta los cambios que hay a lo largo del tiempo en los datos.

Como vemos, estos cuatro problemas son solucionados desde el enfoque de Big Data visto en la sección 3, por lo tanto, es posible construir una arquitectura de Big Data que implemente una solución para prevenir la portabilidad numérica. También, como se mencionó en la subsección 3.1, es importante reconocer cuál de estas 5 V tiene mayor importancia en la arquitectura. Se espera que la arquitectura de Big Data prediga lo más rápido posible qué clientes pretenden portar, por lo tanto la arquitectura debe ser diseñada tomando como premisa principal la velocidad.

5 Una arquitectura de Big Data para solucionar el problema de la portabilidad numérica

Luego de haber visto la relación que existe entre los problemas de la solución clásica y cómo Big Data y las 5 V ofrecen una solución a cada problema, pasamos a explicar los detalles de la arquitectura que se plantea en el presente trabajo.

5.1 Descripción de la solución

La solución propuesta en el presente trabajo aprovechará las nuevas fuentes de información con las que Big Data trabaja, específicamente las de redes sociales, explicadas en la subsección 3.2. El objetivo de 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.

5.2 Componentes de la arquitectura

En la Figura 6 mostramos la arquitectura diseñada en el presente trabajo para poder predecir la portabilidad numérica de clientes. Fue diseñada tomando como referencia la subsección subsección 3.4.

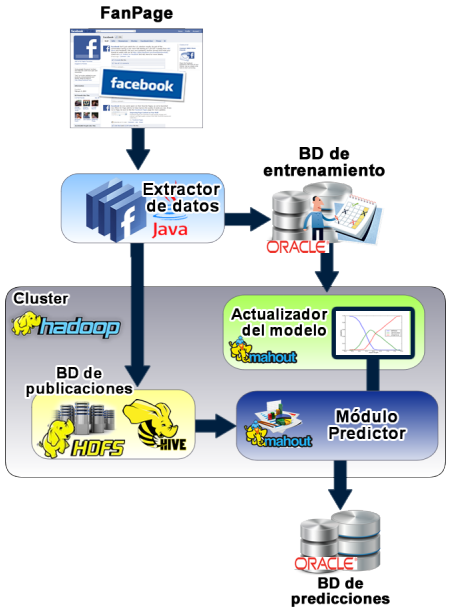

Figura 6: Arquitectura de Big Data para la predicción de la portabilidad numérica

Esta arquitectura está conformada por seis componentes:

El **módulo "extractor de datos"** es el encargado de revisar la página oficial de Facebook de la operadora móvil y descargar las publicaciones y/o comentarios que en ella se realicen. El módulo está encargado de extraer los datos cada hora y almacenarlos en la "base de datos de publicaciones". El módulo extractor de datos también es ejecutado para generar la "base de datos de entrenamiento" y entrenar al modelo. Está implementado en Java y utiliza el API de OpenGraph para conectarse a Facebook.

La **base de datos de publicaciones**, la cual almacena las publicaciones y/o comentarios de Facebook, guarda los datos de manera distribuida por el HDFS de Hadoop y utiliza Hive para gestionarlos, permitiendo trabajar con una sintaxis similar a SQL.

La **base de datos de entrenamiento** es el componente que almacena las publicaciones que servirán para entrenar al modelo de predicción. Cada publicación y/o comentario almacenado aquí debe indicarse si es "negativo", en el caso de opiniones que muestren intenciones de portar, o "positivo", en caso contrario. Esta base de datos está implementada como una base de datos relacional Oracle.

El **módulo "actualizador del modelo"**, es el encargado de generar y actualizar el modelo de

predicción usando como entrada la base de datos de entrenamiento. Está implementado en Mahout para aprovechar el procesamiento paralelo de MapReduce en Hadoop.

El **módulo "predictor"** tiene implementado el modelo de predicción y se encarga de predecir si el comentario de un cliente indica una portabilidad o no. También está implementado en Mahout para aprovechar el procesamiento paralelo de MapReduce en Hadoop.

Finalmente, se encuentra **la base de datos de predicciones** que almacena los resultados obtenidos por el módulo predictor, implementada en Oracle.

5.3 Algoritmo de predicción

El algoritmo de predicción usado en la arquitectura de Big Data es el de "Naive Bayes". Según (Shruti et al., 2014), en el aprendizaje automático, el clasificador Naive Bayes es parte de la familia de los clasificadores de Bayes, pero asume la independencia de las variables y gracias a esto el cálculo de las probabilidades se simplifica. Una ventaja de este clasificador es que requiere una pequeña cantidad de datos de entrenamiento para estimar los parámetros necesarios para la clasificación. Para el cálculo de probabilidades se usan las fórmulas mostradas por (Shruti et al., 2014) adaptándolas a nuestro caso.

$$P\left(\begin{matrix} palabra_i \\ es\ positiva \end{matrix}\right) = \frac{Cantidad\ de\ publicaciones\ positivas\ con\ palabra_i}{Cantidad\ de\ publicaciones\ positivas}$$

$$P\left(\begin{matrix} palabra_i \\ es\ negativa \end{matrix}\right) = \frac{Cantidad\ de\ publicaciones\ negativa\ con\ palabra_i}{Cantidad\ de\ publicaciones\ negativas}$$

Figura 7: Probabilidad de que una palabra sea positiva o negativa

Primero, con las fórmulas mostradas en la Figura 7, evaluamos el conjunto de publicaciones de entrenamiento y obtenemos la probabilidad de que cada palabra sea positiva o negativa.

Luego, con las fórmulas mostradas en la Figura 8, evaluamos todo el conjunto de publicaciones de entrenamiento y calculamos la probabilidad de que el dataset sea positivo o negativo.

$$P\left(\begin{matrix} el\ dataset \\ es\ positivo \end{matrix}\right) = \frac{Cantidad\ de\ publicaciones\ positivas}{Cantidad\ total\ de\ publicaciones}$$

$$P\left(\begin{matrix} el\ dataset \\ es\ negativo \end{matrix}\right) = \frac{Cantidad\ de\ publicaciones\ negativas}{Cantidad\ total\ de\ publicaciones}$$

Figura 8: Probabilidad de que el dataset sea positivo o negativo

Posteriormente, para calcular la probabilidad de que una publicación sea positiva o negativa, dadas las palabras que contienen, usaremos las fórmulas de la Figura 9.

$$P\left(\begin{matrix} publicación\ es\ positiva \mid \\ palabras\ en\ publicación \end{matrix}\right) = \frac{P\left(\begin{matrix} palabras\ en\ publicación \mid \\ publicación\ es\ positiva \end{matrix}\right) x\, P\left(\begin{matrix} el\ dataset \\ es\ positivo \end{matrix}\right)}{P\left(\begin{matrix} palabras \\ en\ publicación \end{matrix}\right)}$$

$$P\left(\begin{matrix} publicación\ es\ negativa \mid \\ palabras\ en\ publicación \end{matrix}\right) = \frac{P\left(\begin{matrix} palabras\ en\ publicación \mid \\ publicación\ es\ negativa \end{matrix}\right) x\, P\left(\begin{matrix} el\ dataset \\ es\ negativo \end{matrix}\right)}{P\left(\begin{matrix} palabras \\ en\ publicación \end{matrix}\right)}$$

Figura 9: Probabilidad de que una publicación sea positiva o negativa

Como "P(palabras en publicación)" es "1", puesto que cada palabra siempre está presente en la publicación, al aplicar Naive Bayes, tenemos las fórmulas mostradas en la Figura 10.

$$P\left(\begin{matrix} publicación\ es\ positiva \mid \\ palabras\ en\ publicación \end{matrix}\right) = P\left(\begin{matrix} el\ dataset \\ es\ positivo \end{matrix}\right) x \prod P\left(\begin{matrix} palabra_i \\ es\ positiva \end{matrix}\right)$$

$$P\left(\begin{matrix} publicación\ es\ negativa \mid \\ palabras\ en\ publicación \end{matrix}\right) = P\left(\begin{matrix} el\ dataset \\ es\ negativo \end{matrix}\right) x \prod P\left(\begin{matrix} palabra_i \\ es\ negativa \end{matrix}\right)$$

Figura 10: Naive Bayes aplicado en las probabilidadesa

Finalmente, para elegir si una publicación es positiva o negativa, se verifica cuál de las dos probabilidades es mayor por medio de la fórmula mostrada en la Figura 11.

$$Clasificación\ de\ publicación = \begin{cases} POSITIVA, si\ P\left(\begin{matrix} publicación\ es\ positiva \mid \\ palabras\ en\ publicación \end{matrix}\right) \geq P\left(\begin{matrix} publicación\ es\ negativa \mid \\ palabras\ en\ publicación \end{matrix}\right) \\ NEGATIVA, si\ P\left(\begin{matrix} publicación\ es\ positiva \mid \\ palabras\ en\ publicación \end{matrix}\right) < P\left(\begin{matrix} publicación\ es\ negativa \mid \\ palabras\ en\ publicación \end{matrix}\right) \end{cases}$$

Figura 11: Clasificación de una publicación

Para mejorar aún más la calidad del clasificador Naive Bayes, se usó el análisis de frecuencias de TF-IDF (Ambele et al., 2014) el cual permite medir la importancia relativa de las palabras y hacer que las palabras menos significativas (stopwords) sean ignoradas en el cálculo de las probabilidades.

El algoritmo clasificador de Naive Bayes fue implementado en Java por medio de la librería Mahout.

5.4 Flujo de predicción

El flujo de predicción que definimos para nuestra arquitectura es el mostrado en la Figura 12.

Primero, los comentarios son extraídos desde la página oficial de Facebook de la operadora móvil y almacenados en la base de datos de publicaciones. Esta extracción se realiza cada hora buscando aquellos comentarios que tengan una fecha

Figura 12: Flujo seguido en la arquitectura para realizar la predicción

posterior a la última extracción realizada. Luego, el módulo predictor analiza todos aquellos comentarios obtenidos con la última extracción y para cada uno predice si es una portabilidad o no. Si la predicción indicara una portabilidad, el comentario es almacenado en la base de datos de predicciones, además, si es la primera vez que un usuario realiza una publicación con intención de portar, los datos del usuario son almacenados.

6 Pruebas realizadas

Para analizar el comportamiento de nuestra arquitectura de Big Data, se realizaron dos tipos de pruebas. La primera de ellas mide el porcentaje de aciertos que tiene el modelo para predecir la portabilidad y la segunda mide la velocidad de respuesta del clúster.

Para la primera prueba se generó el modelo de predicción con Mahout usando unos 10'000 comentarios publicados en las páginas de Facebook de Claro Perú. Cada uno de estos comentarios fue etiquetado como "negativo" en caso de que la opinión indique intención de realizar una portabilidad, y "positivo" en caso contrario. Para etiquetar a un comentario como negativo éste debía cumplir con al menos alguna de las siguientes condiciones: manifestar que algún producto o servicio de la competencia era mejor, manifestar alguna queja sobre un servicio o producto de Claro Perú, manifestar directamente que realizaría una portabilidad, o sugerir una portabilidad a otros clientes. La forma en cómo debía etiquetarse cada comentario como positivo o negativo fue sugerido por analistas de negocio de Claro Perú.

Al ser la página de Facebook de Claro Perú una página orientada al público peruano, el idioma de los comentarios escritos es el español, la lengua más extendida del Perú, sin embargo, según (Shruti et al., 2014) los algoritmos de clasificación de texto son diseñados en su mayoría para trabajar con el idioma inglés y lenguajes europeos, por lo que esta arquitectura también puede ser probada con alguno de estos otros lenguajes.

Con el modelo de predicción generado, se cargaron otros 10'000 comentarios para su validación y se construyó una matriz de confusión para medir su efectividad. En total, para generar el modelo y validarlo se trabajó con 127MB de información.

Para las pruebas de velocidad del clúster, se analizó la mejora en tiempos de respuesta que se obtenía al incrementar el número de nodos en el clúster. Se midieron los tiempos de respuesta del clúster al procesar las 10'000 publicaciones con uno, dos, tres y cuatro nodos.

Todos los nodos usados para las pruebas tenían la misma configuración de software y hardware. Cada uno contaba con un procesador Intel Core i5, 4 GB de RAM, un disco duro rígido de 500 GB y Ubuntu 14.02 con Hadoop 1.2.1.

7 Resultados obtenidos

Los resultados obtenidos al medir el porcentaje de aciertos de publicaciones positivas y negativas fueron calculados a partir de la matriz de confusión generada al realizar las pruebas, la cual es mostrada en la Figura 13.

Predicción / Real	Positivo	Negativo
Positivo	5'839	275
Negativo	722	3'164

Figura 13: Matriz de confusión

Por el lado de las predicciones realizadas para los comentarios positivos, se obtuvo un total de 5'839 aciertos y un total de 722 errores, por lo que

el porcentaje de aciertos para la predicción de comentarios positivos fue de 88.99%.

Por el lado de las predicciones realizadas para los comentarios negativos, se obtuvo un total de 3'164 aciertos y un total de 275 errores, por lo que el porcentaje de aciertos para la predicción de comentarios negativos es de 92.00%.

En conjunto, para los comentarios positivos y negativos, la arquitectura de Big Data propuesta ha obtenido un porcentaje de acierto de un 90.03%.

Los resultados obtenidos en las pruebas del clúster Hadoop, al medir la mejora en tiempos de respuesta agregándole más nodos, son mostrados en la Figura 14.

Nodos	Respuesta (segundos)
1	158
2	100
3	63
4	14

Figura 14: Tiempos de respuesta

Inicialmente, el tiempo de respuesta con un solo nodo era de 158 segundos. Al agregarle otro nodo, el tiempo de respuesta mejora pues disminuye en 58 segundos, al agregarle dos nodos se produce una mejora de 95 segundos menos, y al agregarle tres nodos, el tiempo de respuesta disminuye en 144 segundos.

En la Figura 15, puede verse que al aumentar el número de servidores, los tiempos de respuesta disminuyen de manera lineal.

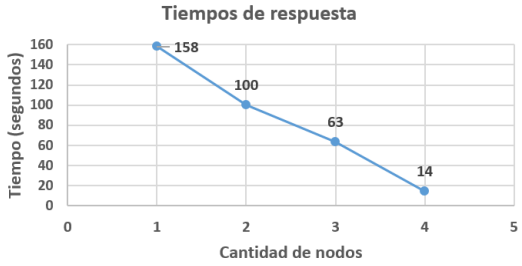

Figura 15: Nodos Vs Tiempos de respuesta

Esta escalabilidad lineal es obtenida gracias a MapReduce, por las razones explicadas en la subsección 3.3

8 Conclusiones y trabajos futuros

En el presente artículo se implementó una arquitectura basada en Big Data la cual permite predecir la portabilidad numérica de clientes en empresas de telefonía móvil. Para ello, se utilizó como fuente de datos los comentarios realizados en la página pública de Facebook de Claro Perú y con ellos se creó un modelo de predicción de análisis de sentimientos en Naive Bayes, el cual obtuvo un alto porcentaje de aciertos (90.03%) y permitió procesar 10'000 comentarios en 14 segundos. La arquitectura permite que la empresa de telefonía móvil pueda identificar en tiempo real qué clientes tienen intención de irse a la competencia, tomando como fuente los datos dejados en redes sociales.

Se comprobó que las 5 V de Big Data se encuentren relacionadas con los problemas planteados en la sección número cuatro del artículo y fueron resueltos, pues las predicciones se realizaron de manera rápida, se trabajó con datos públicos de Facebook y se realizó un proceso en el cual el cliente desconocía el análisis realizado sobre sus comentarios, el proceso fue realizado en tiempo real y sin necesidad de estandarizar los datos, y fue posible hacer seguimiento a la preferencia cambiante del cliente respecto a los servicios que las empresas de telefonía ofrecen.

Por otro lado, (Alan et al., 2015) menciona que los microblogs como Twitter también son utilizados a diario para expresar pensamientos personales en la web, y permite adquirir una valiosa cantidad de opiniones a los investigadores. Como trabajo futuro se propone incluir a Twitter en la arquitectura expuesta como una nueva fuente de datos. Para incluirla será necesario utilizar las API de REST y Streaming que Twitter provee de manera similar al OpenGraph de Facebook visto en el presente artículo.

Para la implementación de la arquitectura, se usaron las tecnologías Hadoop, Mahout y Hive, pero Mike (2013) también propone otras tecnologías como Spark, Storm e Impala que ayudan a mejorar las capacidades de una arquitectura en tiempo real. Como trabajo futuro, estas herramientas de Big Data pueden implementarse en la arquitectura propuesta y podemos llegar a comprobar si llegan a convertirse en una mejor opción tecnológica que solucione el problema.

Por último, es importante destacar que la predicción de la portabilidad numérica juega un rol importante en la industria de la telefonía móvil,

pues con el fin de reducir los diversos costos asociados a la pérdida de clientes, es imperativo que las empresas de telefonía móvil desplieguen modelos predictivos que permitan identificar qué clientes portarán a la competencia. Con estos modelos las empresas podrán formular estrategias de retención de clientes con el objetivo de aumentar sus utilidades y la rentabilidad del negocio.

Referencias

Abdullah Gani, Aisha Siddiqa, Fariza Hanum, y Shahaboddin Shamshirband. 2015. *A survey on indexing techniques for big data: taxonomy and performance evaluation*, Knowledge and Information Systems, vol. 44, n. 2, p. 1-44.

Alan Ritter, Preslav Nakov, Saif Mohammad, Sara Rosenthal, Svetlana Kiritchenko y Veselin Stoyanov. 2015. *SemEval-2015 Task 10: Sentiment Analysis in Twitter*, 9th International Workshop on Semantic Evaluation (SemEval 2015), p. 451-463.

Ambele Robert Mtafya, Dongjun Huang y Gaudence Uwamahoro. 2014. *On Objective Keywords Extraction: Tf-Idf based Forward Words Pruning Algorithm for Keywords Extraction on YouTube*, International Journal of Multimedia and Ubiquitous Engineering, vol. 9, n. 12, p. 97-106.

Carlos Acevedo Miranda, Consuelo V. García Mendoza, Ricardo Clorio Rodriguez y Roberto Zagal Flores. 2014. *Arquitectura Web para análisis de sentimientos en Facebook con enfoque semántico*, Research in Computing Science, n. 75, p. 59-69.

Clement Kirui, Hillary Kirui, Li Hong y Wilson Cheruiyot. 2013. *Predicting Customer Churn in Mobile Telephony Industry Using Probabilistic Classifiers in Data Mining*, International Journal of Computer Science Issues, vol. 10, n. 1, p. 165-172.

Debajyoti Mukhopadhyay, Chetan Agrawal, Devesh Maru, Pooja Yedale y Pranav Gadekar. 2014. *Addressing NameNode Scalability Issue in Hadoop Distributed File System using Cache Approach*, 2014 International Conference on Information Technology, Bhubaneswar, India, p. 321-326.

Dhruva Gajjar. 2014. *Implementing the Naive Bayes classifier in Mahout*, Journal of Emerging Technologies and Innovative Research, vol. 1, n. 6, p 449-454.

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.

G. Bramhaiah Achary, P. Venkateswarlu, y B.V. Srikant. 2015. *Importance of HACE and Hadoop among Big Data Applications*, International Journal of Research, vol. 2, n. 3, p. 266-272.

Kankana Kashyap, Champak Deka y Sandip Rakshit. 2014. *A Review on Big Data, Hadoop and its Impact on Business*, International Journal of Innovatite Research and Development, vol 3, n. 12, p. 78-82.

M. Vasuki, J. Arthi y K. Kayalvizhi. 2014. *Decision Making Using Sentiment Analysis from Twitter*, International Journal of Innovative Research in Computer and Communication Engineering, vol. 2, n. 12, p. 71-77.

Mario Arias, Carlos E. Cuesta, Javier D. Fernández y Miguel A. Martínez-Prieto. 2013. *SOLID: una Arquitectura para la Gestión de Big Semantic Data en Tiempo Real*, XVIII Jornadas de Ingeniería de Software y Bases de Datos, España, p. 8-21.

Mike Barlow. 2013. *Real-Time Big Data Analytics: Emerging Architecture*, O'Really Media.

Organismo Supervisor de Inversion Privada en Telecomunicaciones (OSIPTEL). 2015. *Estado de la portabilidad numérica en el primer trimestre del 2015*, Perú.

Pablo Gamallo, Juan Carlos Pichel, Marcos Garcia, José Manuel Abuín y Tomás Fernández Pena. 2014. *Análisis morfosintáctico y clasificación de entidades nombradas en un entorno Big Data*, Procesamiento del Lenguaje Natural, n. 53, p. 17-24.

Patricio Alfredo Pérez Villanueva. 2014. *Modelo de Predicción de Fuga de Cliente de Telefonía Móvil Post Pago*, Memoria para optar al Título de Ingeniero Civil Industrial. Departamento de Ingeniería Industrial, Universidad de Chile, Chile.

Rakesh Kumar, Neha Gupta, Shilpi Charu, Somya Bansal y Kusum Yadav. 2014. *Comparison of SQL with HiveQL*, International Journal for Research in Technological Studies, vol. 1, n. 9, p. 28-30.

Seyyed Mojtaba Banaei y Hossein Kardan Moghaddam. 2014. *Hadoop and Its Roles in Modern Image Processing*, Open Journal of Marine Science, vol. 4, n. 4, p 239-245.

Shruti Bajaj Mangal y Vishal Goyal. 2014. *Text News Classification System using Naïve Bayes Classifier*, International Journal of Engineering Sciences, vol. 3, p. 209-213.

Sunil B. Mane, Yashwant Sawant, Saif Kazi y Vaibhav Shinde. 2014. *Real Time Sentiment Analysis of Twitter Data Using Hadoop*, International Journal of Computer Science and Information Technologies, vol. 5, n. 3, p. 3098-3100.

Tania Lucía Cobos. 2011. *Y surge el Community Manager*, Razón y Palabra, vol. 16, n. 75.

Yiou Wang, Koji Satake, Takeshi Onishi y Hiroshi Masuichi. 2015. *Improving Churn Prediction with Voice of the Customer*, XXI Annual Meeting Lenguage Processiong Society, p. 816-819.