

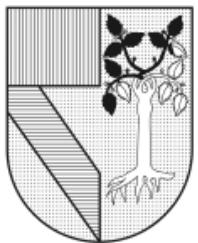
UNIVERSIDAD PANAMERICANA

PREDICCIÓN DEL IMPACTO EN MENSAJES DE TWITTER MEDIANTE REDES NERONALES

MANUEL HONORIO DE LA TORRE RAMÍREZ

Tesis presentada para optar por el grado de
Maestro en Ingeniería con
Reconocimiento de Validez Oficial de Estudios de la
SECRETARÍA DE EDUCACIÓN PÚBLICA,
según acuerdo número 2006098 con fecha 28-II-06.

Zapopan, Jal., Junio del 2015



UNIVERSIDAD PANAMERICANA

Junio del 2015

DR. FRANCISCO ALEJANDRO OROZCO ARGOTE
PRESIDENTE DE LA COMISIÓN DE EXÁMENES DE GRADO
P R E S E N T E .

Me permito hacer de su conocimiento que **MANUEL HONORIO DE LA TORRE RAMÍREZ**, de la Maestría en Ingeniería ha concluido satisfactoriamente su trabajo de titulación con la alternativa Tesis, titulada:

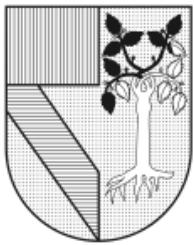
PREDICCIÓN DEL IMPACTO EN MENSAJES DE TWITTER MEDIANTE REDES NERONALES

Manifiesto que, después de haber sido dirigida y revisada previamente, reúne todos los requisitos técnicos para solicitar fecha de Examen de Grado.

Agradezco de antemano la atención prestada y me pongo a sus órdenes para cualquier aclaración.

A T E N T A M E N T E

PhD. ANTONIO VALDERRÁBANO GONZÁLEZ
ASESOR DE TESIS



UNIVERSIDAD PANAMERICANA

DICTAMEN DEL TRABAJO DE TITULACIÓN DE GRADO

SR. MANUEL HONORIO DE LA TORRE RAMÍREZ

P r e s e n t e .

En mi calidad de presidente de la Comisión de Exámenes de Grado, y después de haber analizado el trabajo de titulación presentado por usted en la alternativa de **Tesis** titulada:

PREDICCIÓN DEL IMPACTO EN MENSAJES DE TWITTER MEDIANTE REDES NERONALES

Le manifiesto que reúne los requisitos a que obligan los reglamentos en vigor para ser presentado ante el H. Jurado del Examen de Grado, por lo que deberá de entregar cinco ejemplares como parte de su expediente al solicitar el examen.

ATENTAMENTE

PhD. FRANCISCO ALEJANDRO OROZCO ARGOTE

PRESIDENTE DE LA COMISIÓN

DEDICATORIA

A mis papás: Tey y Manolo, y mis hermanos: Ricky y Monse, como muestra de agradecimiento por su gran apoyo a lo largo de mi vida.

AGRADECIMIENTOS

A Dios, por darme las facultades y condiciones necesarias para permitirme describir un poco de su creación en este trabajo.

A mis maestros a lo largo de todos mis estudios, porque cada uno ha contribuido a formar la persona que soy ahora.

Al profesor Omar Rojas por compartirme sus conocimientos de investigación e impulsarme a escribir esta tesis.

Al profesor Antonio Valderrábano por ser mi director de tesis y apoyarme a finalizar con éxito esta investigación.

RESUMEN

El objetivo de este proyecto de investigación es desarrollar un modelo de predicción, para la cantidad de impacto, medido en clics, *retweets* y favoritos, que va a obtener un mensaje en Twitter (conocido como *tweet*), mediante el uso de una red neuronal artificial.

El modelo tomará como entradas un conjunto de factores que se pueden medir al momento de la publicación de un *tweet*. Estos factores son propuestos de acuerdo a su relevancia en trabajos previos y su correlación con el impacto del mensaje, y se pueden dividir en dos tipos: datos de la cuenta de Twitter desde la que se publicó el mensaje y datos sobre el contenido del mensaje.

Se recolectó una muestra de 46,137 *tweets* que dentro de su contenido tuvieran un *link* (con el objetivo de medir los clics obtenidos). Posteriormente se realizó un modelo de ecuaciones estructurales (PLS-PM) con el fin de analizar la correlación de los factores de entrada en la salida y validar su utilización en el modelo de predicción. Finalmente se construyó la red neuronal artificial, con una arquitectura perceptrón multicapa, que predice el impacto de un mensaje de Twitter; esta red se entrenó utilizando la mitad de los datos recolectados (set de entrenamiento) y una vez lista, se aplicó a la otra mitad de los datos (set de prueba) para validar su correcto funcionamiento y presentar los resultados del modelo.

La innovación de este trabajo radica en que, hasta donde se tiene conocimiento, es el primer modelo que predice clics, *retweets* y favoritos simultáneamente; así como el primer trabajo en incluir los puntajes de influencia de Klout¹ y Moz² (empresas dedicadas al *marketing* en redes sociales) como parte de las entradas del modelo, ya que están estrechamente relacionados con el desempeño histórico de las interacciones que tiene la audiencia de un usuario hacia los mensajes que publica.

¹ <http://klout.com>

² <http://moz.com>

Dentro de las aplicaciones de esta investigación, están el área de *marketing* digital, el estudio de comportamientos humanos en internet y el estudio de la influencia en redes sociales. Mantener la infraestructura tecnológica y humana que requieren las grandes redes sociales cuesta millones de dólares, y muchos se preguntan cómo que es que ofrecen sus servicios gratuitamente; pues bien, lo logran gracias a la publicidad, que está estrechamente relacionado con el “impacto” estudiado en esta investigación. Al lograr predecirlo correctamente, se está prediciendo un mercado que genera billones de dólares al año; he ahí su valor.

ÍNDICE

1. Introducción.....	13
1.1 Objetivo y alcance.....	13
1.2 Trabajo previo	15
1.3 Innovación.....	18
1.4 Contenido.....	18
2. Recolección y filtrado de datos	20
2.1 Introducción.....	20
2.2 Metodología	21
2.3 Medición de clics	21
2.4 Recolección de datos.....	23
2.5 Normalización de datos.....	27
2.6 Conclusión	31
3. Modelación del impacto en mensajes de Twitter	32
3.1 Introducción.....	32
3.2 Metodología	32
3.3 Entrada de datos	32
3.4 Análisis de componentes principales	33
3.5 PLS-PM	38
3.6 Conclusión	41
4. Modelo de predicción.....	43
4.1 Introducción.....	43
4.2 Metodología	45
4.3 Construcción del modelo de predicción	45
4.3.1 Arquitectura de la red neuronal	47
4.3.2 Variables de entrada.....	48
4.4 Optimización del error en la predicción.....	51
4.5 Entrenamiento de la red.....	53
4.6 Resultados	54
4.6.1 Interpretación de los resultados.....	55
4.6.2 Aplicación del modelo	57

4.7 Conclusión	60
5. Conclusiones	61
6. Bibliografía.....	63

Índice de Tablas

Tabla 1. Comparativa de uso entre Goo.gl y Bit.ly en Twitter	22
Tabla 2. Variables capturadas para cada mensaje al momento de su publicación..	25
Tabla 3. Variables capturadas 6 horas después de su publicación.	27
Tabla 4. Conjunto de variables normalizadas que serán utilizadas en el modelo. ...	31
Tabla 5. Resumen de el set de datos utilizado para el análisis.	33
Tabla 6. Resultados para los primeros 2 grupos del PCA.	35
Tabla 7. Variables latentes obtenidas del PCA.	37
Tabla 8. Efecto total entre variables latentes del PLS-PM.	39
Tabla 9. Resultados del modelo exterior del PLS-PM.....	40
Tabla 10. Resultados de crossloadings del PLS-PM.	41
Tabla 11. Experimentos con diferentes arquitecturas de redes neuronales	47
Tabla 12. Equivalencia de entradas en el modelo de predicción	49
Tabla 13. Equivalencia de salidas en el modelo de predicción.....	51
Tabla 14. Resumen de errores en los modelos de predicción.	56
Tabla 15. Datos de entrada obtenidos para el <i>tweet</i> ejemplo.	58
Tabla 16. Datos de impacto obtenidos para el <i>tweet</i> ejemplo.	58
Tabla 17. Ejemplos reales de predicciones.....	59

Índice de Figuras

Figura 1. Gráfica de la Ecuación 1 cuando a_i es igual a 1.....	30
Figura 2. Gráfica de correlaciones en los datos.....	34
Figura 3. Gráfica del análisis de componentes principales.....	35
Figura 4. Ejemplo de un <i>tweet</i> de un usuario verificado y contenido multimedia.	37
Figura 5. Modelo interior del PLS-PM	38
Figura 6. Modelo exterior e interior del PLS-PM	39
Figura 7. Ilustración de una neurona biológica	43
Figura 8. Diagrama de una neurona artificial	44
Figura 9. Ejemplo de un perceptrón multicapa.....	46
Figura 10. Diagrama de la red neuronal.....	48
Figura 11. Gráficas de errores en las predicciones.....	56
Figura 12. Imagen del <i>tweet</i> donde se aplicará la predicción.	57

Índice de Ecuaciones

Eq. 1	29
Eq. 2	30
Eq. 3	31
Eq. 4	49
Eq. 5	49
Eq. 6	49
Eq. 7	50
Eq. 8	50
Eq. 9	50
Eq. 10	50
Eq. 11	50
Eq. 12	51
Eq. 13	52
Eq. 14	52
Eq. 15	53
Eq. 16	53
Eq. 17	54
Eq. 18	54
Eq. 19	54
Eq. 20	54
Eq. 21	55
Eq. 22	55

1. Introducción

1.1 Objetivo y alcance

Twitter³ es una de las redes sociales más importantes a nivel mundial, es un servicio de *micro-blogging*⁴ que permite publicar mensajes cortos, de hasta 140 caracteres. Twitter cuenta con 288 millones de usuarios activos⁵ que lo utilizan para conversar con otros usuarios, compartir noticias o sitios web, publicar sus actividades diarias y obtener información actualizada (Java et al., 2007).

Las publicaciones o mensajes en esta red social reciben el nombre de *tweets*. Cuando un *tweet* es publicado, este es transmitido a los seguidores del autor, es decir, las personas que están suscritas a sus actualizaciones. Los *tweets* aparecen en la página inicial de cada persona, ordenados por orden cronológico descendente.

El objetivo de este trabajo, es desarrollar un modelo que, dadas las variables iniciales de un *tweet* (las características que se pueden mediar al momento exacto de ser publicado), pueda predecir el impacto (medido en clics, *retweets* y favoritos) que tendrá ese mensaje durante su vida útil (el tiempo promedio en que un *tweet* obtiene el mayor número de interacciones; 6 horas según se verá más adelante).

El impacto de un *tweet* se puede interpretar de un sinnúmero de formas, desde las métricas de *social media* básicas: número de impresiones, clics, *retweets*, seguidores, conversiones... hasta métricas sociales y comerciales de gran impacto, que generalmente son imposibles de cuantificar: por ejemplo, en la entrega de los premios Oscar 2014, la conductora Ellen DeGeneres, siendo patrocinada por Samsung, publicó en Twitter una *selfie* que rompió el record como el mensaje más retuiteado en la historia con 3.3 millones de *retweets* y causó que Samsung fuera mencionado en redes más de 900 veces por minuto, causándole un impacto publicitario inmensurable (Vranica, 2014) El *tweet* de Ellen tuvo casi 5 veces más

³ <http://twitter.com>

⁴ <http://en.wikipedia.org/wiki/Microblogging>

⁵ <https://about.twitter.com/company>

retweets que el que tenía el record hasta el momento: “Four more years”⁶, que publicó Barack Obama al ser reelecto presidente de Estados Unidos en 2012. Un ejemplo más reciente fue un *tweet* que realizó Elon Musk (CEO de Tesla) anunciando el lanzamiento de un nuevo producto (en ese momento desconocido); este *tweet* aumentó el valor de su compañía en aproximadamente 1,000 millones de dólares en menos de 10 minutos (Carney, 2015).

Existe un interés particular en Twitter porque es un medio de información constante y actualizado; además tener una política abierta con su contenido, ya que la mayor parte de sus datos son públicos y accesibles desde su API (*Application Programming Interface*; una conexión que permite al acceso a datos y funciones de un sistema), lo que facilita hacer análisis de ellos. Para las marcas es muy útil para conocer en tiempo real la percepción general que tienen las personas sobre sus productos y servicios; además de conocer a sus embajadores de marca en redes sociales, ya que se ha demostrado que mensajes positivos, publicados por las personas correctas pueden incrementar las ventas de una empresa (Dijkman et al., 2013).

Para este trabajo, el impacto a predecir serán las estadísticas de *social media* referentes a la interacción en el mensaje y que se puede obtener de manera pública; y son específicamente tres: clics, *retweets* y favoritos. Se dice que son referentes a la interacción del mensaje porque implican que una persona tuvo interés y realizó una acción específica sobre él, además en caso de que la interacción sea un *retweet*, contribuye a la “viralidad” (propagación) del mensaje, porque permite que más personas tengan acceso a él. Por otro lado se dice que estas variables son públicas porque se puede acceder a ellas mediante una API. Existen otras métricas de impacto en un mensaje de Twitter, como “impresiones” (cantidad de personas que realmente vieron el mensaje), visitas al perfil del usuario, vista de detalles sobre el *tweet*, etc. pero Twitter no provee un acceso público a estas métricas, sino que lo reserva para los anunciantes que contratan sus servicios.

⁶ <https://twitter.com/BarackObama/status/266031293945503744>

Es importante mencionar que esta predicción no será sobre el absoluto de cada una de esta métricas, sino una versión normalizada de las mismas utilizando funciones de proporcionalidad inversa, es decir, más que predecir exactamente cuántos clics, *retweets* y favoritos va a obtener un *tweet*, se quiere predecir un “nivel” de qué tanto impacto obtendrá, donde 0 es nada y cercano a 1 es muy alto.

1.2 Trabajo previo

Se investigó en bases de datos como ArXiv, ResearchGate y Google Scholar sobre predicciones de cualquier métrica de impacto en *tweets* y predicciones de clics en Twitter u otras plataformas para tener un punto de comparación entre ellas. Estas investigaciones relacionadas se describirán en esta sección.

En (Zaman et al., 2010) se realizó uno de los trabajos pioneros de predicción de *retweets* utilizando un modelo probabilístico (*Matchbox*) y utilizando únicamente datos de *retweets* manuales (aquellos que dentro del texto incluyen “RT @”) ya que Twitter no tenía soporte nativo; este modelo tomaba en cuenta los autores del *tweet* y los *retweets* como principales fuentes de información.

Posteriormente (Suh et al., 2010) realizaron un análisis de los factores que se correlacionan con los *retweets*. Encontraron que elementos del contenido como el número de *urls* y el número de *hashtags* de un *tweet*, en conjunto con elementos contextuales como el numero de seguidores, el número de amigos y la antigüedad de la cuenta son factores que se correlacionan efectivamente con la cantidad de *retweets*. Por otro lado, el número histórico de *tweets* que ha publicado un usuario no tiene efecto alguno.

Complementario a ese estudio, (Macskassy & Michelson, 2011) comprobaron que un factor que afecta en gran medida si un *tweet* será retuiteado o no es que el tema del *tweet* tenga relación los intereses descritos en el perfil del usuario. Por su parte, (Metaxas et al., 2013) encontraron que el hacer un *retweet*, indica interés en el mensaje, pero principalmente confianza y empatía con él.

Basados en los trabajos de Zaman y Suh, (Petrovic et al., 2011) realizaron un modelo para predecir para predecir si un *tweet* va a ser retuiteado o no, es decir, no

predicen la cantidad de *retweets* o el impacto que puede obtener un *tweet*, sino sólo dicen si el mensaje va o no a ser retuiteado. Inicialmente comprobaron que esta predicción es posible porque hicieron dos experimentos con humanos, uno mostrándoles sólo el texto del *tweet* y otro además mostrándoles todas las variables relacionadas con el mensaje y el usuario que lo publicó. Habiendo comprobado que la predicción es posible, realizaron un modelo de inteligencia artificial (*passive-aggressive*) que varía conforme la hora del día para realizar la predicción.

Por otro lado, (Peng et al., 2011) estudiaron el comportamiento de los *retweets* con variables similares a otros estudios, pero además analizaron la historia de *retweets* que ha tenido el usuario en mensajes anteriores y encontraron que este es el factor más importante para realizar una buena predicción. En este trabajo no se está realizará este tipo de análisis histórico, sin embargo, se utilizarán los puntajes de Klout y Moz, que dentro de su valor incluyen el desempeño que ha tenido el usuario en mensajes anteriores (especialmente el puntaje de Moz).

Más recientemente, (Zaman et al., 2013) volvió a realizar esfuerzos para construir un modelo de predicción, esta vez utilizando un modelo Bayesiano y obtuvo muy buenos resultados. Una diferencia radical con esta aproximación es que el modelo Bayesiano va actualizando la cantidad de *retweets* que ha tenido el mensaje en ciertos intervalos de tiempo y la predicción se va perfeccionando; mientras que este y la mayoría de otros trabajos realizan la predicción utilizando sólo las variables disponibles al momento de la publicación.

Hasta ahora se ha encontrado que existen varias investigaciones que predicen *retweets*; sin embargo, sólo se encontró un trabajo que predice los clics en esta red social y fue elaborado por ingenieros de Twitter (como se comentará más adelante); además de éste, existen otros trabajos que también predicen los clics, pues es la principal divisa de la publicidad *online*, pero están enfocados principalmente a plataformas con anuncios como Google o Facebook.

(McMahan et al., 2013), ingenieros de Google hicieron un algoritmo que predice el CTR (*click-through rate*, la razón de clics respecto a la cantidad de personas que

vieron el mensaje) de los anuncios que aparecen en su buscador. Combinaron avances teóricos en la materia e ingeniería aplicada para poder lograrlo.

Por otro lado, (Com, 2010) en Microsoft hicieron la misma predicción de CTR en su motor de búsqueda, Bing, pero ellos siguieron un enfoque Bayesiano.

Siguiendo esta misma línea, los ingenieros de Facebook (He et al., 2014) también hicieron una publicación donde utilizaron un algoritmo que utiliza árboles de decisión y regresión logística para mejorar su predicción de clics en los anuncios de Facebook. Mencionan que mejoraron la predicción en un 3% respecto a una que había sido implementada anteriormente.

Similar a este trabajo, (Sanzgiri & Asnani, 2015) construyeron un modelo con redes neuronales y lo combinaron con árboles de decisión para mejorar su desempeño. Lograron un modelo que predice el CTR exitosamente en mensajes patrocinados de motores de búsqueda. Para optimizar la red neuronal utilizaron el método BFGS en lugar de *Gradient-descent* que se usa en este trabajo.

Finalmente y muy acorde a esta investigación, el equipo de desarrollo de Twitter realizó un modelo de predicción utilizando métodos de inteligencia artificial (*machine-learned ranking*) para predecir el CTR en los anuncios de Twitter. Los resultados que obtuvieron fueron buenos, pues mejoraron su predicciones significativamente y lo implementaron con éxito en anuncios reales de Twitter. Es importante señalar dos diferencias importantes con este trabajo: Twitter tiene acceso a muchos datos más de los que hace públicos y los clics no se midieron utilizando Bitly⁷ (que es un servicio de acortamiento de *urls* que ofrece una API con la que es posible obtener estas estadísticas), sino el acortador oficial de Twitter (*t.co*, que tampoco expone públicamente sus datos); ambas condiciones ponen en ventaja a Twitter respecto a cualquier investigador externo.

⁷ <http://bitly.com>

1.3 Innovación

La innovación en este trabajo radica primero en que la predicción se hará, además de *retweets*, para favoritos y clics. La gran mayoría de los trabajos anteriores se centran únicamente en *retweets* como medida de impacto. Además, muy pocos contemplan los favoritos o apenas los mencionan; en opinión del autor, no debería de ser así, porque si bien no proveen tanto impacto y amplificación como los *retweets*, son una medida de aplauso o gusto por el contenido entregado. Hasta nuestro mejor esfuerzo, no se encontró ningún trabajo anterior por parte de investigadores externos a Twitter que haya tratado de predecir los clics en los mensajes de esta red social. Por lo tanto, este debería ser el primer trabajo que modela y predice la combinación de clics, *retweets* y favoritos. Por otro lado, también se innova en el uso de los puntajes de influencia de Klout y Moz como entradas en el modelo de predicción; ya que no se encontró ningún trabajo previo que los utilicen, sino que se limitan a los datos que se pueden obtener directamente de Twitter; estos puntajes brindan información sobre el desempeño que ha tenido el usuario en otros mensajes a lo largo de su historia; por lo tanto son un buen indicio para sus resultados futuros.

1.4 Contenido

En el siguiente capítulo (Recolección y filtrado de datos) se explicará como se recolectaron los mensajes de Twitter y bajo qué criterios se filtraron para utilizarlos como datos de entrenamiento del modelo y prueba. Para poder predecir los clics bajo las mismas condiciones que los *retweets* y favoritos, todos los *tweets* recolectados debieron contener un *link* (un hipervínculo a una página de internet), a pesar de ser un elemento opcional, ya que si no lo tuvieran sería imposible que causaran clics y el muestreo sería dispar. Para la medición de los clics en estos *links* se utilizó Bitly.

En el tercer capítulo (Modelación del impacto en mensajes de Twitter) se validó mediante análisis de componentes principales y PLS-PM que los datos de entrada se correlacionaran con las variables de impacto.

Finalmente, en el cuarto capítulo (Modelo de predicción) se desarrolló el modelo de predicción utilizando una red neuronal de tipo perceptrón con tres capas de

neuronas (descrita más adelante), que de acuerdo a los experimentos tuvo un mejor desempeño. Esta red neuronal se entrenó con el modelo de optimización no lineal *gradient-descent* y el algoritmo *back-propagation*. La recolección de datos se hizo con un programa escrito en JavaScript y se almacenó en una base de datos MySQL. Para conocer los resultados y analizarlos se utilizaron programas escritos en el software estadístico R.

2. Recolección y filtrado de datos

2.1 Introducción

Para este estudio, se quiere predecir el impacto de un mensaje de Twitter (*tweet*). El impacto en un *tweet* se interpretará como la cantidad de clics, *retweets* y favoritos que obtuvo en mensaje por su publicación.

Un *retweet* es un signo de amplificación del mensaje, ocurre cuando una persona lo quiere retransmitir a sus seguidores, manteniendo el autoría del mensaje original. Un favorito es una muestra de aplauso por el mensaje, se produce cuando a una persona le agrada el mensaje y decide dar una señal de ello o guardarlo para sí mismo. Finalmente un clic, es una muestra de interés en el *link* (un hipervínculo a un sitio web) del mensaje y ocurre cuando la persona decide visitar la fuente del contenido que está siendo promovido para verlo completo.

Cualquier mensaje de Twitter está propenso a recibir *retweets* y favoritos por parte de la audiencia que lo recibe, sin embargo, no todos los mensajes pueden recibir clics, ya que no es obligatorio que un *tweet* contenga un *link* en su contenido. Depende de la creatividad y el objetivo del autor.

En general un mensaje de Twitter puede contener los siguientes elementos:

- Texto: Es el contenido principal del mensaje y debe contener un máximo de 140 caracteres.
- Multimedia: Son imágenes y videos embebidos en la parte inferior del mensaje.
- *Hashtags*: Son etiquetas para relacionar el mensaje con un tema en específico. Se caracterizan por tener el símbolo de numeral (#) al inicio y no tener espacios.
- Menciones: Son referencias a un usuario existente en la red social especificado por su nombre de usuario. Se caracterizan por tener una arroba (@) al inicio.
- *Urls*: Son hipervínculos (o *links*) a cualquier sitio web.

- Símbolos financieros: Es la referencia a una compañía, se identifican por el símbolo de dólares (\$) y el símbolo financiero de dicha empresa. El uso de estos elementos es poco común.

2.2 Metodología

Dado que para este estudio se pretende analizar el impacto como producto de los clics, *retweets* y favoritos. Únicamente se tomaron en cuenta mensajes con *links* en su contenido, para garantizar condiciones iguales en los mensajes analizados y que todos tuvieran la posibilidad de recibir clics.

Se realizó una plataforma en NodeJS⁸ (utilizando el lenguaje de programación *JavaScript*) para recibir datos del *stream* (una conexión abierta y en tiempo real) de Twitter, filtrarlos y categorizarlos. La muestra constó de 46,137 *tweets* y se almacenó en una base de datos MySQL para su posterior análisis.

2.3 Medición de clics

La API de Twitter provee información precisa de cada uno de los mensajes que están en la plataforma, entre estos datos está la cantidad total de *retweets* y favoritos que ha obtenido el mensaje hasta ese momento. Sin embargo, no provee nada de información sobre los clics; para resolver este problema se decidió usar una plataforma externa a Twitter que proporcione esta información, dicha plataforma debía cumplir 2 características: tener una API para poder obtener la medición de datos de manera automatizada y ser lo suficientemente popular para obtener una muestra de datos numerosa (es decir, que al menos se redacte 1 *tweet* por segundo a nivel mundial utilizando un *link* de este servicio).

Se consideraron 2 servicios que cumplen con estas características y podían ser utilizados para el estudio:

- Google (goo.gl)⁹: Es el servicio de acortador de *urls* de Google, provee una API precisa y es ampliamente usado en Twitter.

⁸ <https://nodejs.org/>

⁹ <http://goo.gl/>

- Bitly (bit.ly): Es uno de los servicios más populares para acortar *urls* a nivel mundial, precisamente su fama a debe a su extenso uso en Twitter en sus primeros años. Su API es extensa y entrega reportes más completos.

Se hizo un experimento para comparar el uso que se le da en Twitter a ambos servicios, donde las variables a medir fueron: la frecuencia de uso, es decir, cuántos *tweets* lo utilizan por segundo y el porcentaje de mensajes *spam*¹⁰, es decir, de los *tweets* recibidos qué tasa se puede clasificar como indeseable de acuerdo a los criterios mencionados posteriormente en la sección 2.4. El experimento se hizo escuchando el *stream* público de Twitter alternando dos tipos de filtros: “goo gl” para recibir mensajes que utilizaran *links* de Google y “bit ly” para recibir mensajes con *links* de Bitly. Cada *tweet* recibido se contabilizó y se clasificó como mensaje de *spam* o no; al finalizar se calculó la frecuencia de uso y el porcentaje de los mensajes que fueron clasificados como *spam*.

Se hicieron 6 pruebas del experimento: 3 para cada uno los servicios, de manera alternada y con una duración de 10 minutos cada una. Al finalizar las pruebas, se hizo un promedio por cada servicio; los resultados se muestran en la Tabla 1. Cabe destacar que el *stream* de mensajes que se recibió de Twitter, no garantiza que se haya recibido la totalidad de mensajes coincidentes con los parámetros del filtro, ya que Twitter puede limitarlo sin previo aviso. Sin embargo, para Twitter no existe preferencia por ninguno de los servicios, así que se asume que las pruebas ocurrieron bajo condiciones iguales y los resultados no se vieron afectados.

Tabla 1. Comparativa de uso entre Goo.gl y Bit.ly en Twitter

Servicio	Frecuencia de uso	Porcentaje de mensajes <i>spam</i>
Google	10.85 <i>tweets</i> / seg.	24.58%
Bitly	30.53 <i>tweets</i> / seg.	10.08%

Con los datos observados se llegó a la conclusión de utilizar Bitly como proveedor de información de clics para este estudio. Por un lado, su uso en Twitter es aproximadamente 3 veces mayor que el uso del servicio de Google, ya que se

¹⁰ <http://en.wikipedia.org/wiki/Spamming>

publican alrededor de 35 mensajes por segundo (contra 11 de Google). Por otro lado, aunque también existen mensajes *spam* que utilizan Bitly, su porcentaje de uso (10%) es menos de la mitad en comparación con el servicio de Google (25%), pues Bitly es más utilizado para compartir noticias, artículos y promocionales en la red.

2.4 Recolección de datos

Para recolectar *tweets* para el estudio, se utilizó el servicio de *streaming*¹¹ que provee la API de Twitter. Este servicio envía en tiempo real los mensajes que están siendo publicados alrededor del mundo que coinciden con los criterios de búsqueda especificados.

Los mensajes recolectados se acotaron a aquellos que están escritos en un lenguaje dominado por el autor, es decir, inglés o en español.

Concretamente, los criterios de búsqueda utilizados fueron: que estuvieran escritos en inglés o español (*language=es,en*) por los motivos explicados anteriormente y que el mensaje contuviera el texto “bit ly” (*track=bit%20ly*) para asegurar que los mensajes contendrán una *url* acortada por Bitly.

Se comenzaron a recibir *tweets* bajo estos criterios a un ritmo de entre 20 y 40 *tweets* por segundo. Sin embargo, para incrementar la calidad en los datos almacenados y asegurar que los resultados sean confiables, por cada mensaje recibido se aplicó un filtro bajo los siguientes criterios:

- *Link de Bitly*: Se descartaron aquellos mensajes que no contenían al menos una *url* o cuyo dominio sea diferente a <http://bit.ly>. Esto para garantizar que todos los *tweets* eran propensos a obtener clics y ser medidos posteriormente.
- *Spam*: Se filtraron todos los mensajes que tuvieran palabras relacionadas con *spam*, en inglés: (*free, buy, followers, porn...*) o en español (gratis, descuento, sexo...). La lista completa de palabras utilizadas en el filtro puede

¹¹ <https://dev.twitter.com/streaming/reference/post/statuses/filter>

ser requerida al autor en caso de ser necesaria. Este filtro sirve para evitar medir mensajes “basura” (conocido como *spam*), que promueven contenido indeseable.

- *Retweets* manuales: Se filtraron los *tweets* que comienzan con el texto “RT @”; pues significa que no es un mensaje original sino hace referencia a un mensaje del usuario mencionado.
- Fuente del mensaje: Se utilizó la fuente de los mensajes para filtrar únicamente aquellos tweets que fueron publicados desde fuentes oficiales de Twitter (aplicación web, *iOS* o *Android*) y desde los principales clientes no oficiales (*Sprout Social*, *Buffer* y *Hootsuite*) con el fin de evitar capturar mensajes publicados por *bots* (programas que publican mensajes de forma automatizada).
- *Link* nuevo: Los *links* de Bitly puede ser reutilizados y colocados en cualquier parte del internet; esto puede ocasionar clics que no se debieron únicamente al mensaje analizado, sino otros sitios, redes, etc. Esto alteraría gravemente los resultados y resulta indeseable. Para evitar esto, se comprobó que cada *link* fuera nuevo, bajo el criterio de que sólo si no ha obtenido clics hasta ese momento se considera como nuevo.
- Combinación única de *tweet* y *link*: Por cada *link* se aseguró que en ese momento no existiera ningún otro *tweet* que incluyera el mismo *link*. En caso de existir otro con el mismo *link*, ambos se descartan. Este criterio no aplica para *retweets* y se asume que el *stream* proveerá de ambos *tweets* en caso de existir.
- Texto único: Se invalidaron los mensajes cuyo “texto plano” (el texto del mensaje eliminando menciones y *links*) se repite con otro *tweet*. Este criterio evita mensajes de *bots*, *spam* y respuestas automáticas. Para hacer la comparación se hace un *hash* MD5¹² del “texto plano” del mensaje y después es comparado con los existentes, en caso de existir otro igual ambos son descartados.

¹² <http://en.wikipedia.org/wiki/MD5>

Toda vez que un *tweet* llegó por el *stream* y cumplió con estos criterios, se capturaron sus datos relevantes para su posterior análisis. Los datos almacenados se explican en la Tabla 2.

Tabla 2. Variables capturadas para cada mensaje al momento de su publicación.

Variable	Rango	Descripción
userFollowersCount	$[0, \infty)$	La cantidad de seguidores del usuario.
userListedCount	$[0, \infty)$	La cantidad de listas en las que fue añadido el usuario.
userVerified	$\{0,1\}$	Indica si la cuenta del usuario está verificada.
userKloutScore	$[0,100]$	El puntaje de influencia del usuario otorgado por Klout.
userMozScore	$[0,100]$	El puntaje de influencia del usuario otorgado por Moz.
messageReach	$[1, \infty)$	El alcance total que tiene el mensaje al ser publicado.
messageHasMedia	$\{0,1\}$	Indica si el mensaje tiene elementos multimedia.

Es necesario hacer algunas observaciones sobre estos datos:

- *userFollowersCount*: Es la cantidad de seguidores que tiene un usuario en Twitter. Un seguidor es una persona subscrita a las publicaciones de otra persona con el fin de leer lo que escribe. Esta variable no está acotada superiormente porque la cantidad de seguidores puede ser muy alta (en el caso de celebridades, por ejemplo); aunque teóricamente la cantidad máxima de seguidores que puede tener una persona es la población mundial menos uno (porque él mismo no puede ser su seguidor), para hacerlo más simple se asumirá que va de 0 a infinito.
- *userListedCount*: Es la cantidad de listas a las que ha sido añadido un usuario. Una lista es un grupo de usuarios con alguna característica (tema, localización, profesión, edad...) en común, sirven para separar los mensajes de esos usuarios y hacer más fácil su lectura. Frecuentemente las listas se hacen por temas y si un usuario es añadido significa que publica información relevante sobre ese tema; por lo tanto la cantidad de listas es un indicador útil para determinar qué tan interesante es la información que pública alguien para su audiencia.

- *userVerified*: Indica si una cuenta de usuario está verificada por Twitter o no. Generalmente esta opción es solicitada por personas famosas para indicar que su cuenta es la oficial y distinguirla de imitaciones. Es una variable binaria.
- *userKloutScore*: es un puntaje que va de 0 a 100 otorgado por la compañía Klout específicamente para el usuario que publicó el mensaje. Este puntaje proviene de un análisis histórico de la interacción del usuario con su audiencia en todas sus redes sociales o publicaciones. Es una calificación de qué tan influyente es una persona en el internet.
- *userMozScore*: es un puntaje que va de 0 a 100 otorgado por la compañía Moz específicamente para el usuario que publicó el mensaje. Este puntaje es similar al de Klout, pero proviene del análisis de los *retweets* y favoritos que ha tenido este usuario a lo largo de su historia (específicamente en Twitter) y es una referencia del nivel de actividad que tiene su audiencia.
- *messageReach* o alcance del mensaje: es la cantidad de usuarios que están propensos a ver el mensaje. Sólo en caso de que el texto comience un una mención (@) se calcula como el número de menciones a usuarios contenidas en el mensaje; en cualquier otro caso es la cantidad de seguidores que tiene el usuario que hizo la publicación. Esto se debe a la naturaleza de Twitter, ya que si un mensaje comienza un con una mención, supone que es una conversación entre esas personas y debe sólo se le debe mostrar a los interesados. Esta variable no está acotada superiormente porque puede ser igual número de seguidores (que como se vio anteriormente, tampoco está acotada superiormente).
- *messageHasMedia*: es una variable binaria (puede tomar como valores 0 ó 1) que indica si el *tweet* tiene por lo menos una imagen incrustada o un video.

Una vez almacenados estos datos para un mensaje, se le deja de prestar atención hasta transcurridas 6 horas desde su publicación. Un análisis (T. Zaman et al., 2013) reveló que la media de los *retweets* se obtiene entre unos pocos minutos y 3 horas. Por lo tanto, como regla empírica, pero con base en este estudio se determinó esperar 6 horas para medir los resultados obtenidos.

Pasado el tiempo de espera, para cada mensaje se capturaron los datos descritos en la Tabla 3.

Tabla 3. Variables capturadas 6 horas después de su publicación.

Variable	Rango	Descripción
clicksCount	$[0, \infty)$	La cantidad de clics que tuvo el mensaje.
retweetsCount	$[0, \infty)$	La cantidad de <i>retweets</i> que tuvo el mensaje.
favoritesCount	$[0, \infty)$	La cantidad de favoritos que tuvo el mensaje.

Sobre estos datos, es importante señalar que:

- Algunos *tweets* son eliminados durante el tiempo de espera. Estos mensajes fueron descartados inmediatamente de la muestra.
- Adicional al filtro ya tomado para la validación de *links*. La medición de clics se realizó pidiendo a Bitly sólo los resultados provenientes de Twitter (API endpoint: /v3/link/referring_domains), es decir, los que llegaron mediante el dominio <http://t.co> (el utilizado por dicha red social para redireccionar a los visitantes). Esto se realizó para asegurar que los clics provinieron del mensaje en Twitter, ya que en ocasiones el *link* puede llegar a ser publicado en otros sitios (por ejemplo cuando el usuario tiene vinculada su cuenta de Facebook) y esto alteraría los resultados.

Bajo estas condiciones, se almacenaron un total de 143,620 mensajes de Twitter, de los cuales sólo 46,137 cumplieron con las condiciones de los filtros y datos completos. Estos 46,137 mensajes componen la muestra que será analizada en el resto de este estudio.

2.5 Normalización de datos

Las redes neuronales se desempeñan mejor si el valor de las variables de entrada está normalizado (va de 0 a 1) porque que internamente la función de activación trabaja con este rango. Adicionalmente, resulta útil para facilitar la comprensión del impacto que tienen las variables de entrada en las de salida porque se encuentran

en la misma escala. Por lo tanto será necesario asegurar que los datos de entrada cumplan con esta condición.

Dentro de los datos existen algunas variables binarias (que sólo toman valores de 0 o 1), es el caso de *userVerified* o *messageHasMedia* por ejemplo. Sin embargo, se puede observar en la columna “Rango” de la Tabla 2 y la Tabla 3, que hay datos cuyos valores pueden ir desde 0 a infinito. De hecho, dentro de los datos almacenados, existen algunos valores que superan magnitudes de 10^6 (por ejemplo, en el caso de la variable *userFollowersCount*). Dado que el objetivo de este estudio es la elaboración de un modelo de predicción mediante una red neuronal, el rango de los datos capturados es inaceptable y las variables tienen que ser normalizadas para que sus valores tengan un rango admisible.

En total existen 10 variables (7 de entrada y 3 de salida), las cuales se dividieron en tres grupos según el rango de sus valores; y para cada grupo se deberá aplicar un tipo de normalización distinta:

- Sin normalización: Las variables de entrada *userVerified* y *messageHasMedia* son variables binarias (sólo toman valores de 0 ó 1), su rango ya está dentro del necesario y no será necesario aplicar ningún tipo de normalización.
- Normalización para variables acotadas: Para 2 variables de entrada, *userKloutScore* y *userMozScore*, su normalización es sencilla, pues tienen un rango acotado (de 0 a 100). Por lo tanto sólo se deberán dividir entre su rango (100). El resultado significará la fracción obtenida respecto al puntaje máximo posible. Aunque el cálculo de ambos puntajes es privado, internamente tienen una normalización logarítmica o similar; esto se nota fácilmente comparando la influencia real de dos personas respecto a su puntaje. Por ejemplo, el puntaje de Klout de Barack Obama (presidente de Estados Unidos) es de 99, mientras que el puntaje del autor (@manuelmhtr) es de 54; es muy claro que la influencia real de Barack Obama no es casi el doble que la del autor, sino es exponencialmente más grande. El puntaje de Moz actúa de manera muy similar.

- Normalización para variables no acotadas superiormente: 3 de las variables de entrada (*messageReach*, *userFollowersCount* y *userListedCount*) y las 3 variables de salida (*clicksCount*, *retweetsCount* y *favoritesCount*) son sesgadas y tienen un rango que va de 0 a infinito. Se debe aplicar una función de normalización $f_i(x)$ distinta para cada variable i , que sea lineal (para contar con una función inversa que entregue las cifras originales), que evaluada en 0 resulte 0 ($f_i(0) = 0$), y que tenga un máximo de 1 para valores mayores a cero (una asíntota en 1 cuando x tiende a infinito). Esta función $f_i(x)$ será desarrollada más adelante.

Para los datos no acotados superiormente resulta útil la función de proporcionalidad inversa, pues cumple con todas la condiciones de normalización necesarias; sólo basta con hacerla negativa y mover su dominio y rango en una unidad positiva. Adicionalmente, para cada variable distinta se deberá dividir la entrada (x) entre una constante a_i ; cuyo valor dependerá del comportamiento de cada variable y su origen se explicará más adelante. Siguiendo esto, se resolvió que la Eq. 1 es adecuada para utilizarse en la normalización de las variables con rango de 0 a infinito.

$$f_i(x) = 1 - \frac{1}{x/a_i + 1} \quad \text{Eq. 1}$$

La Eq. 1 es útil porque el resultado nos da una idea de la magnitud de la variable en lugar de un número exacto; se relaciona bien con el objetivo de este estudio, ya que no se quiere predecir un número exacto de clics, *retweets* y favoritos, sino un aproximación de su magnitud, donde 0 es nulo y cercano 1 es un valor muy alto. En la Figura 1 se puede apreciar el comportamiento de esta normalización para el ejemplo donde a_i es igual a 1; también es útil para eliminar anomalías en los resultados, pues si un valor es mucho más alto que los demás, al ser normalizado, la diferencia será menor con el resto.

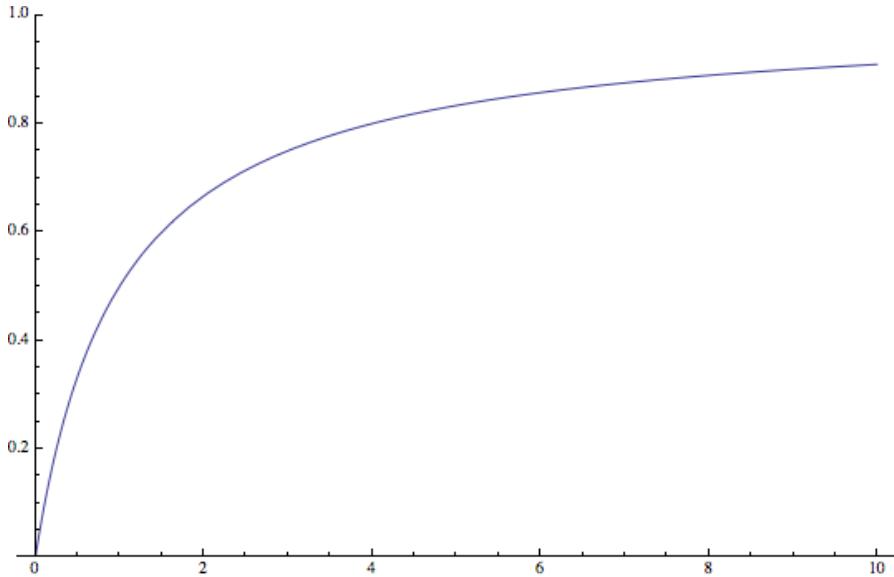


Figura 1. Gráfica de la Eq. 1 cuando a_i es igual a 1. Los valores de a_i dependen de la media y la asimetría estadística (*skewness*) de cada variable. Su cálculo se fundamenta en despejar la Eq. 1 para a_i y trasladar un punto m_i de los datos originales a un punto destino d_i que resulta al ser normalizado, como se muestra en la Eq. 2.

$$a_i = \frac{m_i - d_i m_i}{d_i} \quad \text{Eq. 2}$$

Esta traslación de m_i a d_i debe ser elegida cuidadosamente, de tal forma que los datos normalizados se distribuyan correctamente en su nuevo rango de 0 a 1 y reflejen las propiedades de los datos originales. La media (μ_i) de los datos originales será el punto de origen m_i porque nos indica la tendencia central de los valores, e idealmente debería ser trasladado al punto medio del rango normalizado, es decir, idealmente el punto de destino debería ser $d_i = 0.5$. Sin embargo, la muestra de datos puede estar sesgada positiva o negativamente y se debe mantener ese aspecto después de la normalización. Un indicador que es útil para resolver este problema es la geometría estadística (γ_i o *skewness*), que este caso servirá para calcular la distancia a la que debe estar el punto d_i respecto a 0.5 con el fin de transmitir una geometría similar a la original en los datos normalizados. El rango que

toma γ_i también debe ser normalizado entre los valores de -0.5 y 0.5, pues son los valores máximos que al desplazar el punto d_i desde 0.5 garantizarán que éste permanezca entre 0 y 1. Para normalizar la geometría estadística, se aplicó tangente hiperbólica donde la entrada se suavizó con un valor constante de 100 encontrado empíricamente. De esta forma el cálculo del punto de destino d_i corresponde a la Eq. 3.

$$d_i = 0.5 - 0.5 \frac{e^{\gamma_i/100} - e^{-\gamma_i/100}}{e^{\gamma_i/100} + e^{-\gamma_i/100}} \quad \text{Eq. 3}$$

Siguiendo la Eq. 2 y la Eq. 3 la, se encontraron los valores adecuados de a_i para cada variable en la muestra. Se logró finalmente un conjunto de funciones que se encargarán de normalizar las variables para su posterior uso en la red neuronal y se muestran en la Tabla 4.

Tabla 4. Conjunto de variables normalizadas que serán utilizadas en el modelo.

Variable	Función de normalización
userFollowersRatio	1 - 1 / (userFollowersCount / 33635 + 1)
userListedRatio	1 - 1 / (userListedCount / 315.46 + 1)
userVerified	--
userKloutLevel	userKloutScore / 100
userMozLevel	userMozScore / 100
messageReachRatio	1 - 1 / (messageReach / 33070 + 1)
messageHasMedia	--
clicksRatio	1 - 1 / (clicksCount / 12.78 + 1)
retweetsRatio	1 - 1 / (retweetsCount / 2.27 + 1)
favoritesRatio	1 - 1 / (favoritesCount / 2.05) + 1

2.6 Conclusión

Una vez normalizados los datos, es posible usar este set para hacer un análisis PLS-PM (explicado más adelante) y comprobar que existe correlación entre las variables medibles en la publicación de un mensaje (entrada) y las variables de impacto (salida), y por lo tanto se puede elaborar un modelo para su predicción.

3. Modelación del impacto en mensajes de Twitter

3.1 Introducción

Dadas la variables de entrada capturadas por cada mensaje de Twitter al momento de su publicación, será necesario comprobar que estos datos contribuyen y están correlacionados con el impacto de dicho mensaje (clics, *retweets* y favoritos).

Para realizar esta comprobación se construirá un modelo que involucre todas las variables y la relación que existe entre ellas. Una vez construido el modelo se hará un análisis PLS-PM (*partial least squares path modeling*, explicado posteriormente) para determinar la contribución que tienen entre sí.

Finalmente se analizará la bondad de ajuste que tienen estos datos, para validar que las variables de entrada explican un porcentaje significativo de la varianza en las variables de salida.

3.2 Metodología

Se utilizó un programa escrito en el software estadístico R para realizar la normalización y posteriores análisis de los datos (el código se encuentra disponible en la cuenta de GitHub del autor¹³). Los datos fueron exportados de la base de datos en formato CSV (*comma-separated values*) y en ese formato fueron leídos por el programa en R. Para el análisis de componentes principales se utilizaron las funciones disponibles en la librería estándar *stats* y para realizar PLS-PM se utilizó la librería *plspm*¹⁴ desarrollada por Gastón Sánchez.

3.3 Entrada de datos

El *set* de datos que se utilizó contenía 46,137 mediciones, donde todos los datos cumplieron con las características necesarias para su estudio (explicadas en la sección 2.4). Los datos hasta este momento no estaban normalizados, sino en su estado inicial.

¹³ <https://github.com/manuelmhtr/TweetsImpactPLSPM>

¹⁴ <https://github.com/gastonstat/plspm>

Se aplicó la normalización explicada en la sección 2.5 a los datos recolectados, de tal forma que tuvieran un rango entre 0 y 1. Una vez normalizados, el *set* de datos tuvo las características mostradas en la Tabla 5.

Tabla 5. Resumen de el set de datos utilizado para el análisis.

Variable	Min.	Max.	Mean	Median	Std. dev.	Skewness	Kurtosis
userFollowersRatio	0.0000	0.9937	0.1266	0.0328	0.2093	2.2981	4.6874
userListedRatio	0.0000	0.9918	0.1881	0.0842	0.2330	1.5254	1.4388
userVerified	0.0000	1.0000	0.0462	0.0000	0.2100	4.3206	16.668
userKloutLevel	0.1000	0.9567	0.4605	0.4611	0.1375	-0.1101	0.2499
userMozLevel	0.0100	0.9522	0.3471	0.3501	0.2110	0.1141	-0.7914
messageReachRatio	0.0000	0.9938	0.1247	0.0314	0.2087	2.3187	4.7911
messageHasMedia	0.0000	1.0000	0.2379	0.0000	0.4258	1.2309	-0.4848
clicksRatio	0.0000	0.9976	0.0740	0.0000	0.1684	3.0309	9.5337
retweetsRatio	0.0000	0.9960	0.1331	0.0000	0.2288	1.5660	1.4111
favoritesRatio	0.0000	0.9956	0.1485	0.0000	0.2384	1.4095	0.9402

En la Figura 2 se graficaron las correlaciones existentes en los datos. Se puede apreciar cómo la cantidad de seguidores, el alcance del mensaje y la cantidad de listas en las que aparece el usuario están altamente correlacionadas con el resto de la variables (en promedio 0.48). Por otro lado, en el impacto se notó que los *retweets* y favoritos están altamente correlacionados (0.61); más que con los clics (0.43 y 0.39 respectivamente). Los puntaje de Klout y Moz también tienen una correlación alta (0.84); a pesar de provenir de empresas distintas y tener enfoques diferentes.

3.4 Análisis de componentes principales

El análisis de componentes principales (Wold et al., 1987) o PCA por sus siglas en inglés (*principal component analysis*) es una técnica estadística que dado un conjunto de variables, resalta sus patrones de varianza y es útil para formar grupos de variables correlacionadas. El análisis de componentes principales es una técnica frecuentemente utilizada para construir modelos predictivos; para este estudio, se realizó con el fin de conocer las variables latentes (las variables que no son medidas

directamente sino inferidas a partir de las observables) que se pueden formar a partir de los datos recolectados.

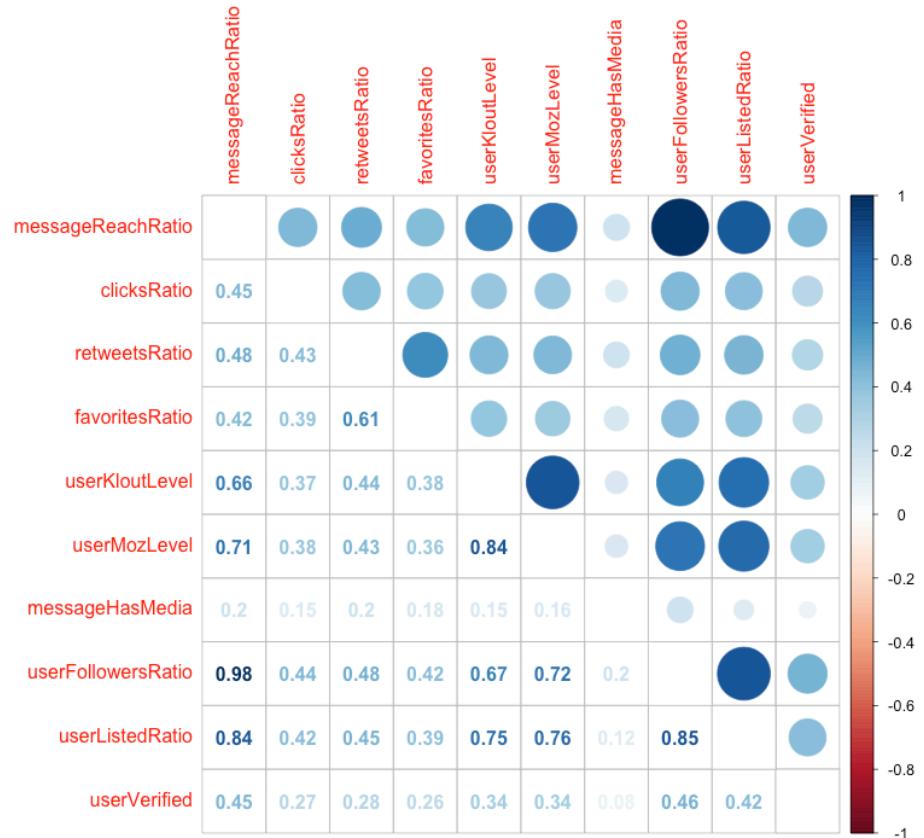


Figura 2. Gráfica de correlaciones en los datos.

Es objetivos de este análisis de componentes principales es conocer las variables latentes que se necesitarán para hacer el análisis PLS-PM y las variables manifiestas (las que fueron observadas) que conforman cada grupo.

Al hacer el análisis de componentes principales y el gráfico de sedimentación (ilustrado en la Figura 3) se puede apreciar que son 2 los componentes principales que explican la mayor varianza en los datos, porque después del segundo punto los valores comienzan a converger. El resto de los grupos, si bien sigue explicando parte de la varianza, no son tan significativos como los primeros 2.

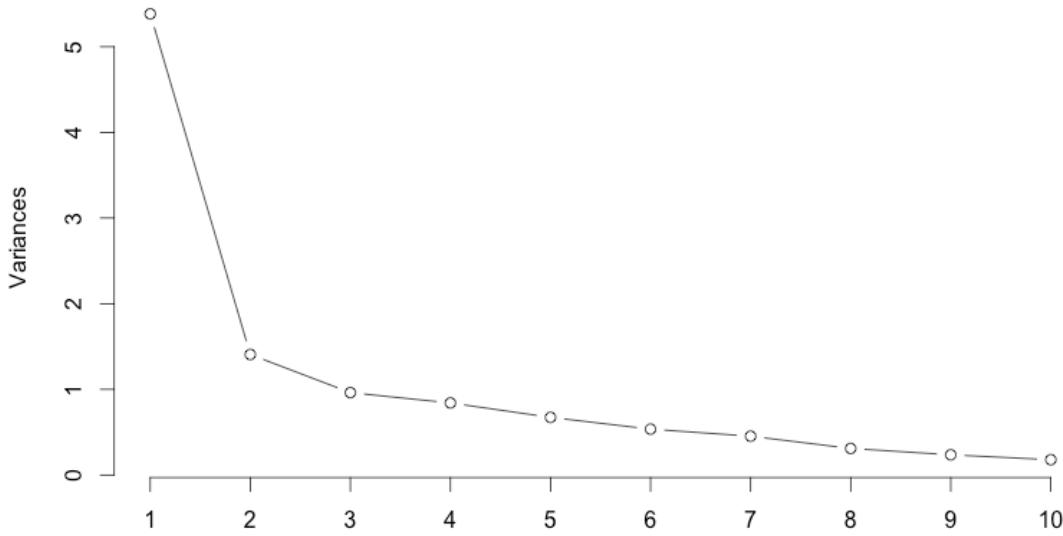


Figura 3. Gráfica del análisis de componentes principales.

En la Tabla 6 se muestran los datos resultantes del análisis de componentes principales para los primeros 2 grupos, que explican la mayor varianza: aproximadamente un 64%. Se resalta en negritas las variables más significativas para cada componente por tener los valores más altos de su grupo (mayores a 0.3).

Tabla 6. Resultados para los primeros 2 grupos del PCA.

Variable	PC1	PC2
messageReachRatio	0.3946	-0.1540
messageHasMedia	0.1108	0.4495
userFollowersRatio	0.3966	-0.1693
userListedRatio	0.3859	-0.2397
userVerified	0.2310	-0.0681
userKloutLevel	0.3574	-0.2025
userMozLevel	0.3661	-0.2346
clicksRatio	0.2522	0.3007
retweetsRatio	0.2845	0.4701
favoritesRatio	0.2570	0.5253

Como se puede observar en los resultados del análisis de componentes principales (Tabla 6), el primer componente está dado por 5 variables: *userFollowersRatio*,

messageReachRatio, *userListedRatio*, *userMozLevel* y *userKloutLevel*. Este grupo tiene perfecto sentido con la realidad porque indica el alcance o nivel de audiencia que tendrá con sus seguidores: las primeras 2 variables son la cantidad de personas propensas a ver el mensaje; *userListedRatio* es la cantidad de listas a las que ha sido añadido el usuario, por lo que entre mayor sea el número de listas de intereses a las que pertenece, es más probable que los mensajes de ese usuario sean más leídos; finalmente los puntajes de Klout y Moz dan una idea de qué tan probable es que esa audiencia tenga alguna interacción con el mensaje. Comprendido esto, este grupo se relaciona con la audiencia del autor y se llamará *Audience*.

El segundo grupo se conforma de los tres parámetros de impacto en un mensaje (*favoritesRatio*, *retweetsRatio* y *clicksRatio*) y la variable *messageHasMedia*. Para que coincida con el objetivo de estudio, el grupo se conformará sólo por las variables de impacto y se descartará *messageHasMedia*. De esta forma el grupo cobra sentido, pues es el resultado al que se quiere llegar y el hecho de que se encuentren en el mismo componente nos indica que los tres tipos de impacto están estrechamente relacionados, es decir, en la medida que la cantidad de uno aumenta, es probable que también lo haga en las otras en su respectiva proporción. Este grupo indica el impacto que obtuvo el mensaje y será llamado *Impact*.

Las variables restantes (*messageHasMedia* y *userVerified*) conformarán el tercer grupo. Aunque este grupo no está tan definido como los dos anteriores, pues no destacó en ninguno de los componentes principales, se utilizará porque ambos factores hacen sentido con la realidad, pues afectan la apariencia del mensaje al ser publicado en Twitter. El primero, *messageHasMedia*, indica si el mensaje tiene una imagen o video en él, es decir, agrega contenido multimedia debajo del mensaje; esto generalmente afecta la interacción con el mensaje pues consigue en promedio un 35% más *retweets* (Chen et al., 2013). El otro factor *userVerified*, indica si la propiedad de la cuenta fue verificada por Twitter; y afecta al mensaje agregando un ícono azul de verificación al lado del perfil del usuario; esto ocurre principalmente con cuentas de celebridades o instituciones y le dan más confianza al usuario, afectando positivamente su interacción. En la Figura 4 se ilustra un *tweet* con ambos factores. Este grupo nos indica variantes el mostrar el contenido del mensaje, por lo que se llamará *MessageContent*.

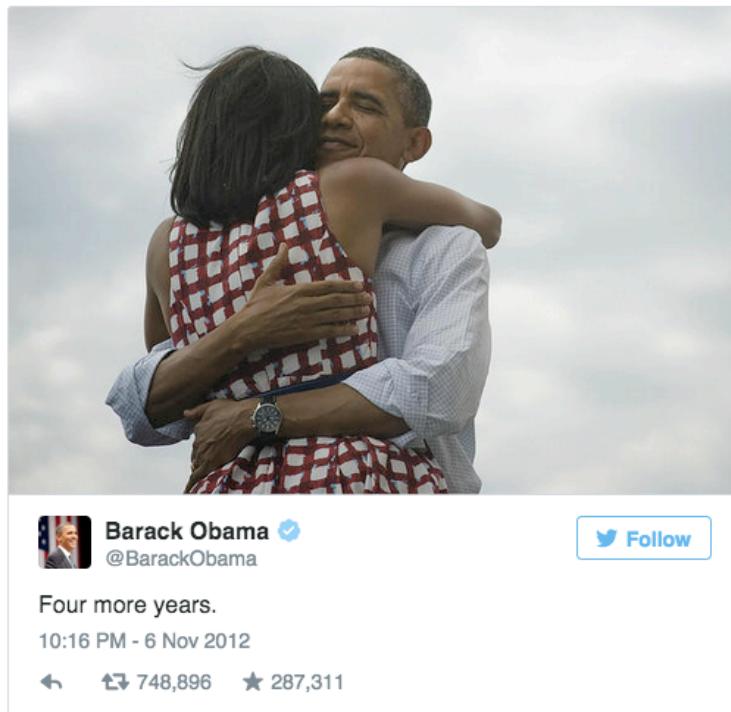


Figura 4. Ejemplo de un *tweet* de un usuario verificado y contenido multimedia.

Realizado el PCA, se clasificaron los datos de entrada en tres variables latentes: *Audience*, *Impact* y *MessageContent*, mostradas en la Tabla 7.

Tabla 7. Variables latentes obtenidas del PCA.

Audience	Impact	MessageContent
messageReachRatio	clicksRatio	messageHasMedia
userFollowersRatio	retweetsRatio	userVerified
userListedRatio	favoritesRatio	--
userKloutLevel	--	--
userMozLevel	--	--

3.5 PLS-PM

PLS-PM (*partial least squares path modeling*) es una técnica para modelar ecuaciones estructurales, que son un método estadístico para estimar la relación causal entre variables. Se utiliza para conocer la cantidad de varianza explicada en una relación de variables y probar un modelo teórico de causalidad. Para este estudio, el PLS-PM servirá para verificar que los datos de entrada y de impacto (variables de salida) están relacionadas y comprobar que es posible construir un modelo de predicción. Para realizar esta comprobación se tomarán en cuenta los pesos de impacto entre variables y la bondad de ajuste del modelo, como se verá a continuación.

El modelo interior (el modelo de relación entre variables latentes) se construyó a partir de las variables latentes definidas en la sección 3.4 mediante PCA, donde el impacto es resultado de los valores de audiencia y contenido del mensaje. De esta manera el modelo interior se representa como en el diagrama en la Figura 5.

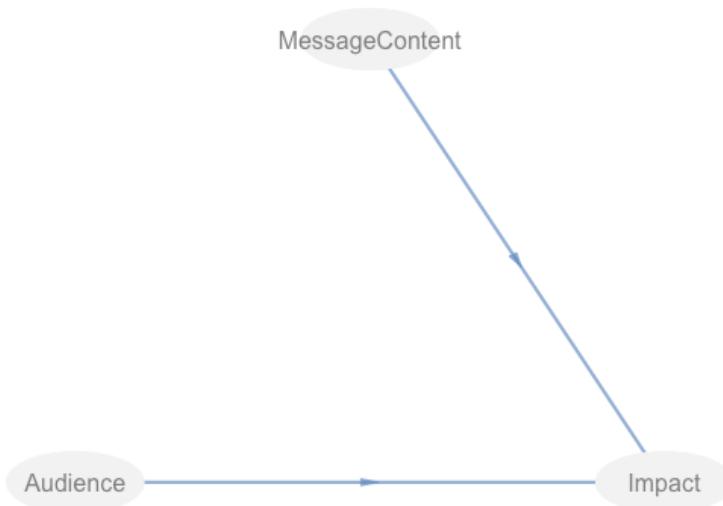


Figura 5. Modelo interior del PLS-PM.

Para construir el modelo exterior (el modelo de relación que incluye las variables manifiestas u observadas), se utilizaron las variables manifiestas agrupadas de acuerdo al análisis de componentes principales y con una relación reflexiva (las

mediciones son causadas por el fenómeno inferido) a su variable latente. El modelo final (interior y exterior) resulta como se muestra en la Figura 6.

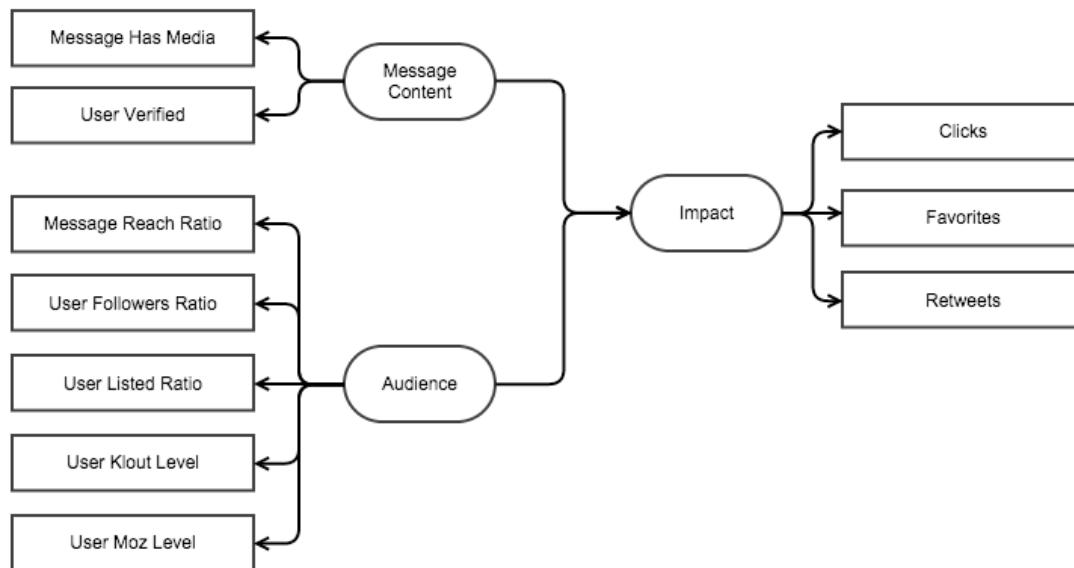


Figura 6. Modelo exterior e interior del PLS-PM.

Siguiendo este modelo, se realizó el PLS-PM en el software R y se obtuvo una bondad de ajuste (*GOF*) de 0.5014; y un efecto entre variables latentes listado en la Tabla 8.

Tabla 8. Efecto total entre variables latentes del PLS-PM.

Relation	Direct	Indirect	Total
MessageContent -> Audience	0.000	0.000	0.000
MessageContent -> Impact	0.155	0.000	0.155
Audience -> Impact	0.506	0.000	0.506

Los resultados obtenidos para el modelo exterior, se muestran en la Tabla 9, y demuestran que los datos sí están correlacionados y cada uno de ellos tiene un aporte y un peso significativo en el modelo propuesto; esto es, porque las cargas entre las variables tienen puntajes elevados (mayores a 0.2), mientras que la communalidad (un factor que indica qué tan preciso es el ajuste de los datos respecto a su correlación) es mayor a 0.7 en todas la variables excepto en

messageHasMedia (0.348), *clicksRatio* (0.550) y *favoritesRatio* (0.673); sin embargo se decidió mantenerlas como parte del modelo. *messageHasMedia* debe permanecer, porque como se comentó anteriormente (Chen et al., 2013) comprobaron que los *tweets* con imágenes consiguen aproximadamente 35% más *retweets*; por su parte *clicksRatio* y *favoritesRatio* también debe permanecer pues son parte de las variables que se van a predecir y son parte indispensable de este estudio, además, su comunalidad se acerca mucho al valor mínimo deseado (0.70).

Tabla 9. Resultados del modelo exterior del PLS-PM.

	Weight	Loading	Communality	Redundancy
MessageContent				
1 messageHasMedia	0.525	0.590	0.348	0.000
1 userVerified	0.810	0.852	0.727	0.000
Audience				
2 messageReachRatio	0.236	0.931	0.867	0.000
2 userKloutLever	0.208	0.856	0.732	0.000
2 userMozLever	0.204	0.885	0.782	0.000
2 userFollowersRatio	0.233	0.938	0.879	0.000
2 userListedRatio	0.219	0.926	0.857	0.000
Impact				
3 clicksRatio	0.400	0.742	0.550	0.193
3 retweetsRatio	0.445	0.856	0.734	0.258
3 favoritesRatio	0.393	0.820	0.673	0.237

Para validar el modelo resultante del análisis de componentes principales, se verificó en los *crossloadings* (cargas cruzadas) que el puntaje que tiene cada variable fuera mayor en el componente que le fue asignado que en cualquier otro; de no ser así significaría que esa variable manifiesta debería ser colocada en otro componente. Los *crossloadings* se listan en la Tabla 10 y se puede verificar que en todos los casos se cumple correctamente la relación.

Tabla 10. Resultados de crossloadings del PLS-PM.

		MessageContent	Audience	Impact
MessageContent				
1 messageHasMedia		0.590	0.185	0.218
1 userVerified		0.852	0.445	0.336
Audience				
2 messageReachRatio	0.471		0.931	0.560
2 userKloutLever	0.355		0.856	0.493
2 userMozLever	0.357		0.885	0.485
2 userFollowersRatio	0.481		0.938	0.553
2 userListedRatio	0.401		0.926	0.521
Impact				
3 clicksRatio	0.299		0.454	0.742
3 retweetsRatio	0.333		0.504	0.856
3 favoritesRatio	0.302		0.436	0.820

Para finalizar el análisis PLS-PM, se realizó una prueba de *bootstrapping*, para comprobar que el modelo funciona aún cuando existen cambios en los datos y no está sobre ajustado (*over-fitting*) a los datos iniciales.

3.6 Conclusión

A pesar de que la bondad del ajuste de los datos no es mayor a 0.70, como la literatura lo sugiere, sino de 0.5014, el modelo resultante es aceptable porque todas las variables tienen pesos y cargas significativas que además afectan positivamente al resultado. La pruebas de *bootstrapping* confirmaron que el modelo sigue funcionando con datos diferentes y el análisis de cargas cruzadas demostró que la agrupación que se hizo de variables es la correcta.

Una bondad de ajuste de 0.5014 indica que el modelo predice un 25% de la varianza. Esto se debe a que existe mucha aleatoriedad en los datos; también se debe a que en la realidad, mucho del impacto en un mensaje depende el contenido lingüístico que tiene ya que al final, las personas que leen el mensaje lee dan un sentido de acuerdo a sus vivencias y es en ese momento donde le dan relevancia o

no. Sin embargo esto es algo mucho más subjetivo y difícil de medir, por lo que no se involucró en este estudio pero podría extenderse una línea de investigación hacia esta rama.

Se pudo comprobar que los datos están correlaciones entre sí. Por lo tanto se pueden utilizar estas estos datos como entradas en un red neuronal artificial y construir un modelo de predicción.

4. Modelo de predicción.

4.1 Introducción

Para desarrollar un modelo de predicción del impacto que va a tener un mensaje de Twitter se utilizará una red neuronal artificial. Las redes neuronales artificiales (McCulloch & Pitts, 1943) son una herramienta utilizada para resolver problemas de *machine learning* o inteligencia artificial, son un algoritmo de aprendizaje continuo que en cada iteración puede ser entrenado para mejorar su desempeño y son capaces de resolver problemas con múltiples entradas cuyo efecto en la salida es desconocido.

Las redes neuronales artificiales están inspiradas en el comportamiento de las redes neuronales biológicas. Las neuronas biológicas (Figura 7) son las células que conforman esta red. Una neurona puede recibir señales eléctricas de múltiples neuronas en sus terminales, llamadas dendritas (*dendrites*); si la intensidad de estas señales de entrada es suficientemente fuerte (sobrepasa el umbral de la neurona) entonces se “activa” y envía una señal eléctrica mediante su axón a otras neuronas que igualmente pueden activarse o no.

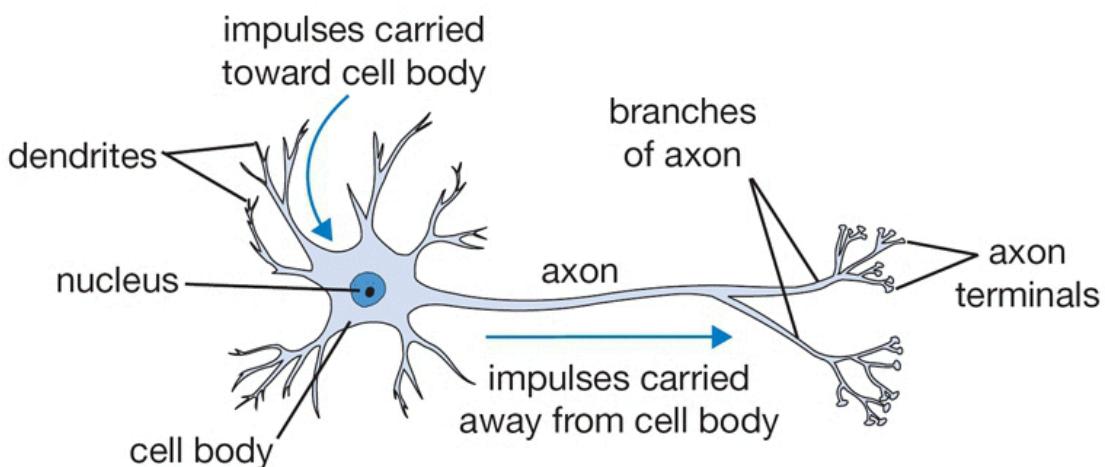


Figura 7. Ilustración de una neurona biológica.

En el caso de las redes neuronales artificiales, su unidad básica también es una neurona (artificial), ilustrada en la Figura 8, y su estructura es similar a la biológica. Cuenta con múltiples entradas, donde cada una es multiplicada por un peso distinto.

En el “cuerpo” de la neurona estas entradas se suman y a este resultado se le aplica una función de activación. La función de activación puede ser cualquier función cuya salida se encuentra entre 0 y 1. La función de activación más simple (similar a la biológica) es una comparación, donde si la entrada supera una constante k la salida será 1, y en caso de ser menor, la salida será 0. Para este trabajo, la función de activación elegida es la función sigmoide (detallada más adelante) pues al tomar un rango continuo e infinito de valores (en lugar de sólo dos valores: 0 ó 1), conserva información de la entrada y ocasiona que la red neuronal tenga un mejor desempeño. Finalmente la salida de la función de activación se puede conectar a la entrada de otra neurona artificial.

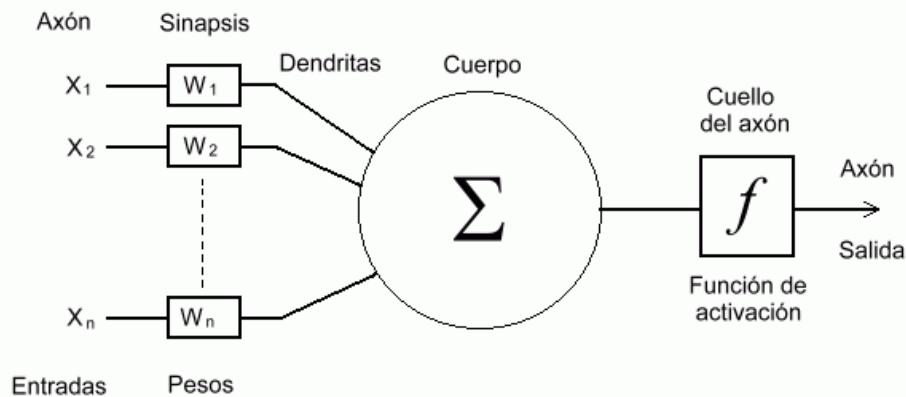


Figura 8. Diagrama de una neurona artificial.

Las redes neuronales tienen una multitud de aplicaciones; por ejemplo, son muy útiles en problemas de clasificación: para realizar algoritmos de reconocimiento de escritura (Graves, 2012), que está siendo usado en los teclados de teléfonos inteligentes; para clasificar imágenes según su contenido (Krizhevsky, et al. 2012); o para reconocimiento facial (Lawrence, et al. 1997) que es ampliamente usando en redes sociales y software de fotografía. También se utilizan mucho para hacer modelos de predicción, en el área de finanzas por ejemplo, para predecir el costo de las acciones (Wu, 2015); o en el sector de telecomunicaciones, Google utiliza redes neuronales para predecir el consumo de sus recursos y optimizar la eficiencia energética en sus centros de datos (Gao, 2013).

Dada la versatilidad de las redes neuronales artificiales, se eligieron como el método adecuado para resolver este problema de predicción de impacto en mensajes de Twitter, que tiene una naturaleza estocástica y no lineal.

4.2 Metodología

Se realizó un programa escrito en JavaScript (corriendo sobre NodeJS), conectado a una base de datos de MySQL que leyó de manera aleatoria (utilizando la función RAND¹⁵ que implementa MySQL) la mitad de los datos recolectados y sobre ellos se iteró n veces para entrenar la red neuronal artificial (donde $n = 10,000$, pues se encontró que el error converge y es inútil realizar más iteraciones). Una vez entrenada la red, se leyeron los datos recolectados restantes, es decir, aquellos que no se utilizaron en la etapa de entrenamiento, y por cada uno se realizó la predicción, se calculó su error respecto a los resultados reales (suma de los resultados reales menos la suma de las predicciones) y finalmente se guardaron todos los errores obtenidos en la misma base de datos.

Para construir la red neuronal en JavaScript se utilizó la librería Synaptic¹⁶ escrita por Juan Cazala, cuyo algoritmo de entrenamiento se fundamenta en (Monner & Reggia, 2012).

El código de este programa está en un repositorio privado, pero puede ser compartido a petición de los interesados.

Posterior al entrenamiento y cálculo de los errores en la predicción, se realizó un programa en el software estadístico R, que se conecta a la misma base de datos donde ha sido almacenada toda la información para su posterior análisis y presentación de los resultados.

4.3 Construcción del modelo de predicción

Se hicieron diversos experimentos de entrenamiento con la finalidad de encontrar empíricamente la mejor arquitectura de red neuronal artificial para ser utilizada como

¹⁵ https://dev.mysql.com/doc/refman/5.1/en/mathematical-functions.html#function_rand

¹⁶ <https://github.com/cazala/synaptic>

modelo de predicción. Como punto de partida para los experimentos se utilizó una arquitectura de tipo perceptrón multicapa (MLP por sus siglas en inglés). Una arquitectura de perceptrón multicapa consiste en múltiples capas de neuronas, donde cada capa está completamente conectada a la siguiente y la información siempre viaja en una sola dirección (*feedforward*). Un perceptrón se compone de una capa en entrada (*input*), h capas ocultas (*hidden*) y una capa de salida (*output*). En la Figura 9 se ejemplifica un perceptrón multicapa que tiene 4 entradas, 1 capa oculta compuesta de 4 neuronas y 3 salidas.

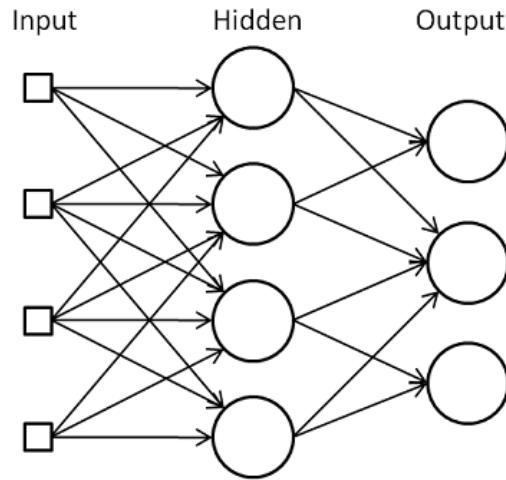


Figura 9. Ejemplo de un percetrón multicapa.

Para realizar los experimentos se seleccionó una muestra aleatoria de la mitad de los datos recolectados y se iteró 1,000 veces con un ritmo de entrenamiento (r) de 0.1. Aunque con 1,000 iteraciones en la etapa de entrenamiento el error no logra converger completamente, el objetivo de esta etapa no es llegar al modelo final de predicción, sino únicamente comparar entre las diferentes arquitecturas posibles; por lo tanto, para disminuir el tiempo de cada prueba se iteró la muestra menos veces. El ritmo de aprendizaje, se explicará a detalle posteriormente, por el momento sólo es necesario mencionar que es una constante con un rango entre 0 y 1 que afectará en la velocidad de aprendizaje de la red neuronal, un valor alto podría hacer que el error nunca converja, mientras que un valor bajo podría hacer que el modelo se detenga en un mínimo local, en vez del mínimo global que se busca.

Bajo estas condiciones, se hicieron múltiples pruebas, variando la cantidad de neuronas (n) en la capa oculta en incrementos de 5 hasta llegar a 20. Se encontró que el menor error se consigue con 5 neuronas en la capa oculta. Posteriormente se hicieron pruebas alrededor de ese resultado (desde 3 hasta 7 neuronas); pero con 5 neuronas el desempeño siguió siendo mejor. Hecho esto, se prosiguió variando en escala logarítmica el ritmo de aprendizaje, desde 0.001 hasta 1 y se descubrió que el mejor desempeño se consigue con un ritmo de aprendizaje (r) de 0.01. Los resultados obtenidos se muestran en la Tabla 11.

Tabla 11. Experimentos con diferentes arquitecturas de redes neuronales

Arquitectura	Error (MSE)	Tiempo de ejecución
Perceptrón ($n = 5, r = 0.1$)	0.2508	122.86
Perceptrón ($n = 10, r = 0.1$)	0.2581	130.38
Perceptrón ($n = 15, r = 0.1$)	0.2712	138.17
Perceptrón ($n = 20, r = 0.1$)	0.2825	142.41
Perceptrón ($n = 3, r = 0.1$)	0.2515	120.10
Perceptrón ($n = 4, r = 0.1$)	0.2516	123.82
Perceptrón ($n = 6, r = 0.1$)	0.2526	122.17
Perceptrón ($n = 7, r = 0.1$)	0.2520	124.71
Perceptrón ($n = 5, r = 0.01$)	0.2437	123.23
Perceptrón ($n = 5, r = 0.001$)	0.2470	124.31
Perceptrón ($n = 5, r = 1$)	0.3611	124.62

Como se puede observar la arquitectura que tuvo mejores resultados es el perceptrón de una capa oculta con 5 neuronas y un ritmo de aprendizaje de 0.01; por lo tanto se eligió para construir el modelo de predicción de este estudio.

4.3.1 Arquitectura de la red neuronal

Se encontró que la red neuronal artificial adecuada para realizar el modelado de la predicción es de tipo perceptrón multicapa (también conocido como *feedforward*), específicamente esta red está compuesta por 3 capas de neuronas: una capa de entradas (*input layer*) formada de 7 neuronas para las entradas más una neurona

con un valor de 1 para simular el error; una capa oculta (*hidden layer*) con 5 neuronas más una para simular el error; y finalmente una capa de salidas (*output layer*) compuesta de 3 neuronas, cuyo valor será la predicción de los clics, *retweets* y favoritos respectivamente. De esta forma, la red neuronal se construyó como se ilustra en la Figura 10.

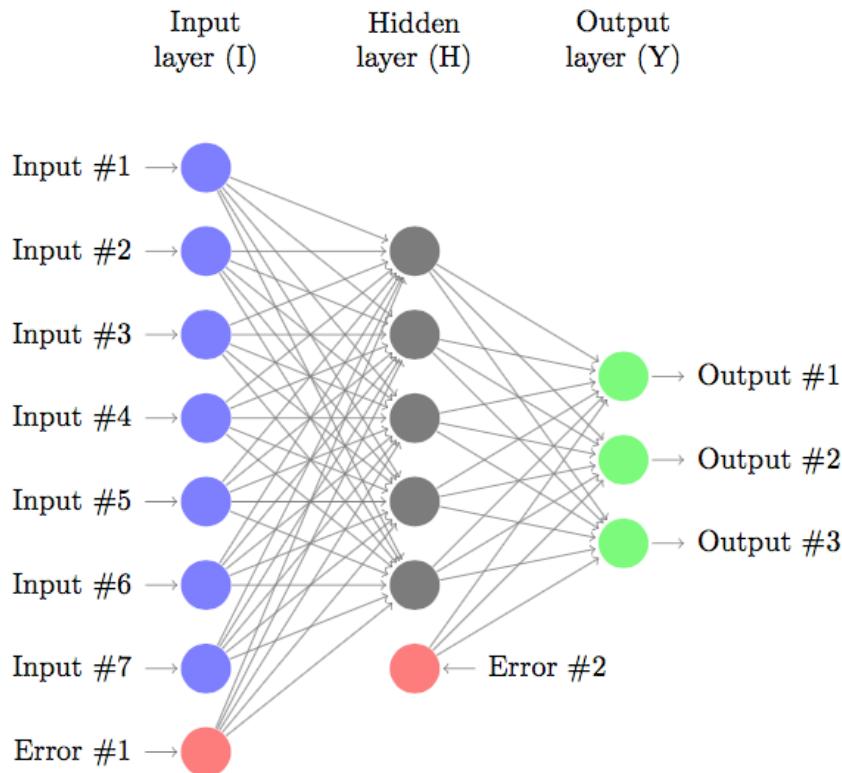


Figura 10. Diagrama de la red neuronal.

4.3.2 Variables de entrada

Derivado de los resultados obtenidos en el análisis PLS-PM se seleccionaron como entradas las 7 variables anteriormente propuestas en la Tabla 2 y como salidas se utilizarán las variables medidas del impacto en los mensajes de Twitter (*clicksRatio*, *retweetsRatio* y *favoritesRatio*) listadas en la Tabla 3. Ambas, entradas y salidas, tienen una magnitud que va de 0 a 1.

La arquitectura de esta red da lugar a 17 neuronas: 7 de entrada ($x_1 - x_7$, en la capa *input*), 2 de error (θ_1 y θ_2 , en las capas *input* y *hidden* respectivamente), 5 ocultas

$(h_1 - h_5$, en la capa *hidden*) y 3 de salida ($o_1 - o_3$, en la capa *output*); y 58 pesos ($\omega_{1,1} - \omega_{8,5}$, que relacionan la capa *input* con la capa *hidden* y $\mathcal{W}_{1,1} - \mathcal{W}_{6,3}$, que relacionan la capa *hidden* con la capa *output*) como se aprecia en la Figura 10.

De acuerdo a esto, se definirá un vector x (Ecuación 4) que corresponde a las variables de entradas, cuya equivalencia se define en la Tabla 12. Este vector x será parte de la capa *input* (I) de la red neuronal.

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_7 \end{bmatrix} \quad \text{Eq. 4}$$

Tabla 12. Equivalencia de entradas en el modelo de predicción

Entrada del modelo	Variable de entrada
x_1	messageReachRatio
x_2	messageHasMedia
x_3	userFollowersRatio
x_4	userListedRatio
x_5	userVerified
x_6	userKloutLevel
x_7	userMozLevel

Los pesos ω , se pueden definir por la Ecuación 5 mientras que los pesos \mathcal{W} por la Ecuación 6.

$$\omega = \begin{bmatrix} \omega_{1,1} & \cdots & \omega_{1,5} \\ \vdots & \ddots & \vdots \\ \omega_{8,1} & \cdots & \omega_{8,5} \end{bmatrix} \quad \text{Eq. 5}$$

$$\mathcal{W} = \begin{bmatrix} \mathcal{W}_{1,1} & \cdots & \mathcal{W}_{1,3} \\ \vdots & \ddots & \vdots \\ \mathcal{W}_{6,1} & \cdots & \mathcal{W}_{6,3} \end{bmatrix} \quad \text{Eq. 6}$$

La función de activación de cada neurona será la función sigmoide (Ecuación 7) cuya derivada (Ecuación 8) será necesaria más adelante para optimizar el modelo.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad \text{Eq. 7}$$

$$\sigma'(x) = \frac{1}{1 + e^{-x}} (1 - \frac{1}{1 + e^{-x}}) \quad \text{Eq. 8}$$

La primera etapa produce la capa oculta (\hat{H}) que cuenta con 6 neuronas; ocurre al multiplicar la capa de entrada (I) por la transpuesta de los pesos ω , aplicar la función de activación y añadirle el error 2 (θ_2). Esta primera etapa del modelo se explica en la Ecuación 9.

$$\hat{H} = \begin{bmatrix} \hat{H}_1 \\ \vdots \\ \hat{H}_6 \end{bmatrix} = \begin{bmatrix} \sigma(\omega^t \begin{bmatrix} x \\ \theta_1 \end{bmatrix}) \\ \theta_2 \end{bmatrix} \quad \text{Eq. 9}$$

A continuación, para calcular los valores en la capa de salida (\hat{Y}), basta con hacer la multiplicación matricial de la capa oculta (\hat{H}) por la transpuesta de los pesos \mathcal{W} y aplicar la función de activación (σ) como se demuestra en la Ecuación 10.

$$\hat{Y} = \begin{bmatrix} \hat{Y}_1 \\ \hat{Y}_2 \\ \hat{Y}_3 \end{bmatrix} = \sigma(\mathcal{W}^t \hat{H}) \quad \text{Eq. 10}$$

De esta forma las 3 salidas: la predicción de impacto (clics, *retweets* y favoritos), en los mensajes de Twitter estará dada en su forma más simplificada por la Ecuación 11.

$$\hat{Y} = \sigma \left(\mathcal{W}^t \begin{bmatrix} \sigma(\omega^t \begin{bmatrix} x \\ \theta_1 \end{bmatrix}) \\ \theta_2 \end{bmatrix} \right) \quad \text{Eq. 11}$$

Donde el vector \hat{Y} , corresponde a las variables de salida listadas en la Tabla 13.

Tabla 13. Equivalencia de salidas en el modelo de predicción

Entrada del modelo	Variable de entrada
\hat{Y}_1	clicksRatio
\hat{Y}_2	retweetsRatio
\hat{Y}_3	favoritesRatio

4.4 Optimización del error en la predicción

Los pesos ω y \mathcal{W} inicialmente son elegidos aleatoriamente, sin embargo, la meta de una red neuronal artificial es “aprender” con cada ejecución e ir ajustando estos pesos a su valor óptimo.

El objetivo de esta red neuronal es que las predicciones de los clics, *retweets* y favoritos sean lo más cercanas a los valores medidos. Por lo tanto, la diferencia entre los valores reales y los valores que genera el modelo, es indeseable y se le conocerá como costo (E). Este costo significa el error que tiene la red al predecir los resultados.

El costo del modelo de predicción puede ser cualquier función (E) que exprese la diferencia entre los valores predichos (\hat{Y}_k) y los valores reales de clics (Y_1), *retweets* (Y_2) y favoritos (Y_3); y que al ser minimizada se encuentren los valores ideales de los pesos (ω y \mathcal{W}), pues al tender a 0 significará que la predicción es igual al resultado real. Por lo tanto, se eligió la función de costo (E) expresada en la Ecuación 12.

$$E = \frac{1}{2} \sum_{k=1}^3 (Y_k - \hat{Y}(\omega, \mathcal{W})_k)^2 \quad \text{Eq. 12}$$

La función elegida eleva al cuadrado la diferencia entre los valores reales y los obtenidos en la predicción, para hacerlos positivos y crear una función convexa, esto será de utilidad al momento de optimizarla por el método de *gradient descent*, como se explicará más adelante. La constante 1/2 que aparece en la ecuación sirve únicamente para facilitar los cálculos posteriores y no afecta al resultado pues es sólo un escalar.

Comprendido esto, el entrenamiento de la red neuronal se puede definir como el siguiente problema de optimización:

Variables de decisión: ω, \mathcal{W} (58 variables)

Función objetivo: $\min E = \frac{1}{2} \sum_{k=1}^3 (Y_k - \hat{Y}(\omega, \mathcal{W})_k)^2$

Dado que existen 58 variables de decisión, ajustar dichas variables hasta encontrar la solución óptima tomaría demasiado tiempo como para ser viable. Por lo tanto se seguirá el método *gradient descent*, que consiste en encontrar el gradiente (la dirección donde se maximiza la razón de cambio o derivada) de la función y ajustar las variables de decisión en dirección negativa para encontrar su mínimo.

Como parte del modelo de minimización del error, las derivadas parciales de los pesos \mathcal{W} están dadas por la Ecuación 13 y para los pesos ω por la Ecuación 14.

$$\frac{\partial \hat{Y}_k}{\partial \mathcal{W}_{j,k}} = \delta_k \hat{H}_j = \hat{Y}_k (1 - \hat{Y}_k) \hat{H}_j \quad \text{Eq. 13}$$

$$\frac{\partial \hat{Y}_k}{\partial \omega_{i,j}} = \delta_j x_j = x_j \hat{H}_j (1 - \hat{H}_j) \sum_{k=1}^3 \delta_k \mathcal{W}_{j,k} \quad \text{Eq. 14}$$

Una vez conocida la tasa de mayor cambio de la función (gradiente) para cada uno de los pesos, estos se deben modificar en sentido contrario, pues se busca encontrar el mínimo de la función. Por ende se logra que la red “aprenda” y mejore su predicción. Sin embargo, si el cambio se hace de manera acelerada, es posible que nunca se llegue a un mínimo, porque el gradiente no mantiene la misma dirección indefinidamente y se puede pasar, por otro lado, si se hace a un ritmo muy lento se puede detener en un mínimo local y la predicción no llegaría a su mejor desempeño. Para controlar esta razón de cambio, el gradiente se multiplica por un factor r , llamado “ritmo de aprendizaje” que reduce la tasa de cambio y otorga un mayor control durante la etapa de entrenamiento de la red. Incluyendo esta

constante, se determina que la tasa de cambio para los pesos $\mathcal{W}_{j,k}$ ($\Delta\mathcal{W}_{j,k}$) debe ser como se indica en la Ecuación 15 y la tasa de cambio en los pesos $\omega_{i,j}$ ($\Delta\omega_{i,j}$) según la Ecuación 16.

$$\Delta\mathcal{W}_{j,k} = -r \frac{\partial \hat{Y}_k}{\partial \mathcal{W}_{j,k}} \quad \text{Eq. 15}$$

$$\Delta\omega_{i,j} = -r \frac{\partial \hat{Y}_k}{\partial \omega_{i,j}} \quad \text{Eq. 16}$$

El ritmo de aprendizaje r toma valores entre 0 y 1. Para el modelo desarrollado se eligió un valor r de 0.01 debido a los experimentos que se mostraron anteriormente en la sección 4.2.

Dadas las ecuaciones anteriores se tiene un modelo que en cada predicción minimiza su error local y se acerca a su mínimo global; se dice que el modelo se está entrenando.

4.5 Entrenamiento de la red

Para entrenar la red neuronal, primero se construyó el modelo de la red según se ha descrito anteriormente (perceptrón multicapa con 7 entradas y 3 salidas). Una vez construida, se leyó el set de datos recolectados y se dividió aleatoriamente en dos partes: una mitad de entrenamiento y otra mitad de prueba.

Por cada elemento en el set de datos de entrenamiento se realizó una predicción, se comparó con los resultados reales y se retroalimentó con el error a la red neuronal para optimizar sus pesos con un ritmo de aprendizaje de 0.01. Este procedimiento se repitió 10,000 veces (para que el error llegara a su convergencia) para todos los datos en el set de entrenamiento y hasta ese punto se consideró que la red estaba entrenada, es decir, se obtuvo el modelo de predicción.

Una vez que la red fue entrenada, se leyó el set de datos de prueba y por cada entrada se hizo una predicción utilizando el modelo obtenido, se comparó con los

resultados reales y se almacenó el error (descrito en la Ecuación 12) en la base de datos; en esta etapa la red neuronal no se entrenó de nuevo (no hubo retroalimentación) para no modificar el modelo. Hecho esto, se obtuvieron los resultados de la predicción listos para ser analizados.

4.6 Resultados

Después de entrenar la red neuronal utilizando el set de datos de entrenamiento, y haber realizado las predicciones del set de datos de prueba, se encontraron los pesos ω y \mathcal{W} que optimizan el modelo de predicción. Los resultados para ω se muestran en la Ecuación 17 y los resultados para \mathcal{W} en la Ecuación 18:

$$\omega = \begin{bmatrix} -6.2892 & -6.2383 & 0.4127 & -5.9092 & -9.1534 \\ -0.7212 & -0.03555 & -0.4563 & 3.4079 & 0.4844 \\ 5.8872 & 1.1250 & -2.6496 & 9.5305 & 13.0609 \\ -2.9336 & 7.0617 & 0.7075 & -9.2989 & -4.0825 \\ 0.0908 & -1.5017 & -0.9758 & -7.0215 & 0.2640 \\ -4.1182 & 13.3808 & 4.7098 & 3.4871 & -22.4583 \\ -0.3710 & -16.9508 & -8.3701 & 2.4187 & 10.4445 \\ 7.5439 & 6.1984 & 2.0456 & -5.6804 & 5.3226 \end{bmatrix} \quad \text{Eq. 17}$$

$$\mathcal{W} = \begin{bmatrix} -2.4001 & -3.2700 & -4.9342 \\ -2.4486 & -2.7466 & -2.8393 \\ -2.5982 & -2.2133 & -1.3258 \\ 0.5955 & 1.1122 & 0.9147 \\ -2.1673 & -2.3904 & -1.5077 \\ 3.5429 & 5.7010 & 6.8438 \end{bmatrix} \quad \text{Eq. 18}$$

Aplicando estos resultado al modelo de optimización propuesto en la Ecuación 12, se encuentra que el error mínimo es de 2,851.25 (Ecuación 19).

$$E = 2851.25 \quad \text{Eq. 19}$$

Finalmente, se llegó a un modelo de predicción de impacto en mensajes de Twitter, que se expresa en la Ecuación 20:

$$\hat{Y}(x) = \sigma \left(\mathcal{W}^t \left[\begin{matrix} \sigma(\omega^t [x]) \\ 1 \end{matrix} \right] \right) \quad \text{Eq. 20}$$

4.6.1 Interpretación de los resultados

Analizando los pesos ω del modelo de predicción se puede apreciar que las filas 3, 6 y 7 tienen valores absolutos más grandes; esto significa, que las entradas x_1 , x_2 y x_3 (cuya equivalencia son las variables *userFollowersRatio*, *userKloutScore* y *userMozScore* respectivamente) tienen mayor relevancia en el modelo.

Después de realizar la predicción sobre todo el set de datos de prueba, se almacenaron los errores de cada entrada y al analizarlos se obtuvieron resultados positivos.

Para tener un punto de referencia que sirva para comprar el modelo de predicción y comprender el desempeño de la red neuronal, se realizó una predicción de control. Esta predicción de control fue la media para cada variable de impacto de los resultados reales, que previamente se habían almacenado. Esta predicción de control \bar{Y}_c se puede expresar como lo indica la Ecuación 21 y su resultado es un vector de tres constantes, es decir, una predicción para cada variable de impacto.

$$\bar{Y}_c = \frac{1}{n} \sum_{i=1}^n \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix}_n \quad \text{Eq. 21}$$

Aplicando esta predicción de control al set de datos de prueba, se obtiene la predicción mostrada en la Ecuación 22 para las tres variables de salida (*clicksRatio*, *retweetsRatio* y *favoritesRatio*).

$$\bar{Y}_c = \begin{bmatrix} 0.0744 \\ 0.1342 \\ 0.1506 \end{bmatrix} \quad \text{Eq. 22}$$

Teniendo el modelo de predicción propuesto y el de control, se prosiguió a comprar sus errores respecto a los valores reales. Para ello se calcularon los errores MAD (*mean absolute deviation*), MSE (*mean squared error*), el costo del modelo (E , Ecuación 19) y error cuadrado máximo. Los resultados de estos errores para cada predicción se muestran en la Tabla 14.

Tabla 14. Resumen de errores en los modelos de predicción.

Valor	MAD	MSE	E	Error ² Max
Predicción de control	0.4999	0.3515	4054.72	6.5364
Predicción de la red neuronal	0.3958	0.2472	2851.25	5.3052
ΔE (Incremento del error)	-20.82%	-29.67%	-29.68%	-18.83%

Al calcular los errores se puede apreciar que el error disminuye en más de un 18% en todos los casos, y para el MSE la disminución del error disminuye en casi un 30%. Esto demuestra que el modelo obtenido es mejor y se acerca a la predicción real del impacto en los tweets.

Otra observación de interés son las gráficas de los errores elevados al cuadrado para ambas predicciones, pues a simple vista se nota una disminución sustancial en los errores del modelo obtenido a partir de la red neuronal. En la Figura 11 se muestran los errores de ambas predicciones, a la izquierda se muestran los errores de la predicción de control, y a la derecha se muestran los errores de la predicción de la red neuronal.

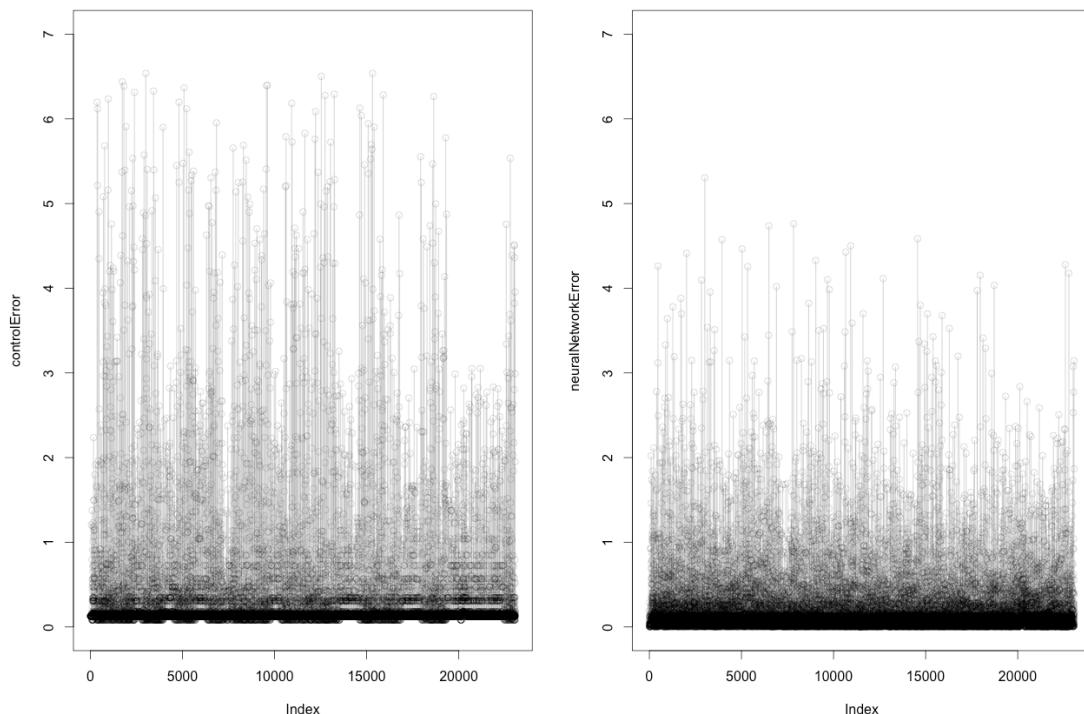


Figura 11. Gráficas de errores en las predicciones

4.6.2 Aplicación del modelo

Una vez construido el modelo de predicción y habiendo comprobado de funciona, se prosiguió a observar su desempeño para un caso en particular. Se eligió un *tweet*¹⁷ del usuario @luxury_travel publicado el 7 de abril de 2015. Este mensaje está escrito en inglés y es una recomendación a una publicación de un *blog* de viajes que habla de cuatro recorridos turísticos que se pueden hacer desde París. El texto es “4 amazing mini-trip ideas from Paris <http://bit.ly/1DYZc4>” y también incluye una imagen del Valle del Loira en Francia; el *tweet* completo se ilustra en la Figura 12.



Figura 12. Imagen del *tweet* donde se aplicará la predicción.

Al momento de su publicación (7 de abril de 2015 a las 12:14pm CDT), el mensaje se obtuvo a través del *stream* de Twitter, se validó y se capturaron sus datos. En la Tabla 15 se muestran los datos que se almacenaron para este mensaje directamente de Twitter y su valor al ser normalizados.

¹⁷ https://twitter.com/luxury_travel/status/585490724642250752

Tabla 15. Datos de entrada obtenidos para el *tweet* ejemplo.

Variable	Valor medido	Valor normalizado
messageReachRatio	488,436	0.9366
messageHasMedia	1.00	1.0000
userFollowersRatio	488,435	0.9356
userListedRatio	7,347	0.9588
userVerified	0.00	0.0000
userKloutLevel	79.81	0.7981
userMozLevel	76.65	0.7665

Se puede observar que el usuario cuenta con muchos seguidores (488,435) y está en varias listas de usuarios (7,347), por lo que se puede decir que el usuario es popular. Los puntajes de Klout y Moz lo confirman, pues tiene 79.81 y 76.65 respectivamente (de un máximo de 100.00). Intuitivamente se puede decir que el impacto que tendrá su mensaje será alto, es decir, las variables normalizadas de salida tendrán un valor cercano a 1.

Al aplicar el modelo de predicción (Ecuación 20) para este mensaje en particular, se obtuvo un valor de 0.4168 para *clicksRatio* (\hat{Y}_1), 0.7550 para *retweetsRatio* (\hat{Y}_2) y 0.8009 para *favoritesRatio* (\hat{Y}_3). Al finalizar el tiempo de espera (6 horas) a partir de la publicación se midieron los resultados de impacto de este mensaje y se encontró que obtuvo 12 clics, 7 *retweets* y 12 favoritos; que al ser normalizados equivalen a 0.4842 para *clicksRatio* (Y_1) 0.7551 para *retweetsRatio* (Y_2) y 0.8540 para *favoritesRatio* (Y_3). Los valores de la predicción y los valores reales se muestran en la Tabla 16 para su comparación.

Tabla 16. Datos de impacto obtenidos para el *tweet* ejemplo.

Variable de impacto	Predicción (\hat{Y}_x)	Medición (Y_x)	Diferencia
clicksRatio	0.4168	0.4842	-0.0674
retweetsRatio	0.7550	0.7551	-0.0001
favoritesRatio	0.8009	0.8540	-0.0531

Finalmente, como se mostró en la Ecuación 12, se determinó que el “costo” (error) de esta predicción tiene una magnitud de 0.0037. Este valor es cercano a cero, por lo que se puede decir que la predicción fue exitosa y demuestra de nuevo que el modelo funciona.

Para reforzar la validez del modelo, en la Tabla 17 se presentan ejemplos adicionales de predicciones en mensajes de Twitter. En la tabla se muestran los datos reales de algunos mensajes recolectados con su respectiva predicción. Como se ha visto en gráficos anteriores, una gran parte de los *tweets* tienen nulos o muy pocos resultados de impacto, esto puede dar paso a soluciones triviales cercanas a cero; sin embargo, para esta muestra de resultados, se eligieron intencionalmente mensajes con impacto medio o superior para demostrar que aún para mensajes con impacto variado el modelo funciona y puede predecir buenos resultados.

Tabla 17. Ejemplos reales de predicciones.

Id	Impacto real			Predicción del impacto			
	Clicks (Y_1)	Retweets (Y_2)	Favoritos (Y_3)	Clicks (\hat{Y}_1)	Retweets (\hat{Y}_2)	Favoritos (\hat{Y}_3)	Costo (E)
1	0.1353	0.3058	0.3279	0.1315	0.3457	0.3294	0.0008
2	0.1901	0.4684	0.4938	0.1812	0.4959	0.4708	0.0007
3	0.1353	0.3058	0.3279	0.1500	0.3284	0.3052	0.0006
4	0.2384	0.4684	0.4938	0.2264	0.4976	0.4709	0.0008
5	0.1353	0.3058	0.3279	0.1320	0.3048	0.2428	0.0036
6	0.1353	0.3058	0.3279	0.1287	0.3703	0.3093	0.0023
7	0.1353	0.3058	0.3279	0.1577	0.3541	0.3036	0.0017
8	0.1353	0.4684	0.4938	0.1992	0.4531	0.4775	0.0023
9	0.2384	0.4684	0.4938	0.2350	0.4864	0.4160	0.0032
10	0.0726	0.3058	0.3279	0.1030	0.2851	0.2723	0.0022

Con este trabajo se quería lograr un modelo que partiendo de las variables y la información disponible para un mensaje de Twitter (*tweet*) al momento de su publicación, predijera el impacto que éste iba a tener después de un tiempo medio de vida (6 horas); donde el impacto se mide en la cantidad de clics, *retweets* y

favoritos obtenidos. Con los resultados obtenidos hasta ahora queda demostrado que el enfoque que se siguió fue correcta y se logró obtener un modelo que aún con ciertas limitaciones predice ese impacto.

4.7 Conclusión

Después de varias iteraciones se llegó a una arquitectura de red neuronal artificial de tipo perceptrón con 3 capas y un total de 17 neuronas que predice el impacto de mensaje de Twitter con un error mínimo.

Se demostró matemáticamente cómo funciona este modelo y se realizó un modelo de optimización no lineal que minimiza el error en cada predicción. También se realizó un programa de cómputo que toma los datos recolectados previamente, utiliza la mitad para entrenar la red neuronal y con la otra mitad realiza predicciones para conocer el desempeño de la red.

Finalmente mediante diversas aproximaciones se demostró la efectividad de la red respecto a otra predicción de control y se enlistaron algunos ejemplos de mensajes reales donde el modelo fue utilizado. Aunque la predicción no es cien por ciento precisa, cumple con su trabajo y se acerca al resultado real, a pesar de ser un proceso tan estocástico por su naturaleza social.

5. Conclusiones

Se realizó un modelo para predecir el impacto (medido en clics, *retweets* y favoritos) en mensajes de Twitter, usando como entradas únicamente variables disponibles al momento de la publicación. El modelo de predicción se realizó utilizando una red neuronal de tipo perceptrón con 3 capas: una con 7 entradas, una capa oculta de 5 neuronas y una capa de 3 salidas.

Para construir el set de datos se realizó un programa de cómputo que recoleta mensajes de Twitter con características adecuadas, así como información pública de otras API (Bitly, Klout y Moz). Estos datos se normalizaron, procesaron y almacenaron en una base de datos para su posterior análisis.

Para determinar que las variables de entrada se correlacionan con el impacto, se realizó un análisis de ecuaciones estructurales (PLS-PM) y se seleccionaron 7 variables que se pueden medir al momento de la publicación de un *tweet*. Estas variables se clasificaron en dos tipos: las que se relacionan con el contenido del mensaje y las que se relacionan con el usuario que lo publicó.

Posteriormente se construyó la red neuronal y se entrenó con la mitad de los datos recolectados, para después hacer la predicción utilizando la otra mitad de los datos y conocer el desempeño del modelo propuesto.

Se obtuvieron resultados satisfactorios con el modelo de predicción construido, pues el error (*MSE*) se logró disminuir en casi un 30% respecto una predicción de control (donde la predicción es igual a la media de los resultados reales), lo que demuestra la efectividad de la red neuronal.

Aunque los resultados fueron buenos, desde el análisis de ecuaciones estructurales se notó que gran parte de la varianza en los datos no se explica por las variables de entrada. Esto puede ser una base para continuar esta línea de investigación, incluyendo factores que no han sido contemplados en este trabajo. El análisis histórico por usuario podría ser un factor, que apoyado de los puntajes de Klout y Moz, puede ser muy poderoso para conseguir una predicción más precisa, ya que

es muy probable que un usuario tenga resultados similares a los que ha tenido anteriormente. Por otro lado, puede que también sea útil integrar variables del entorno, por ejemplo, la actividad en los seguidores del usuario o la hora del día en la que ocurre la publicación, ya que el impacto en un mensaje se correlaciona bastante con la cantidad de personas que tienen acceso a él.

Se debe tomar en cuenta que al ser un proceso estocástico, porque depende de reacciones sociales y emociones humanas, predecirlo se vuelve complicado; sería necesario desarrollar inteligencia artificial que realice un análisis léxico y semántico del contenido del mensaje para comprender su contenido y pueda determinar que tan atractivo es para una persona; ese podría ser el inicio de nueva investigación.

6. Bibliografía

- Carney, M. (2015). Elon Musk created nearly \$1B in value today with a single tweet. Retrieved April 9, 2015, from <http://pando.com/2015/03/30/elon-musk-created-nearly-1b-in-value-today-with-a-single-tweet/>
- Chen, T., Salaheldeen, H. M., He, X., Kan, M., & Lu, D. (n.d.). VELDA: Relating an Image Tweet 's Text and Images.
- Com, R. M. (2010). Web-Scale Bayesian Click-Through Rate Prediction for Sponsored Search Advertising in Microsoft ' s Bing Search Engine. *Search*, (April 2009), 13–20. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.165.5644&rep=rep1&type=pdf>
- Dijkman, R., Ipeirotis, P., Aertsen, F., & Helden, R. Van. (2013). USING TWITTER TO PREDICT SALES : A CASE STUDY, 1–14.
- Gao, J. (2013). Machine Learning Applications for Data Center Optimization. *Google White Paper*, 1–13.
- Graves, A. (2012). Offline arabic handwriting recognition with multidimensional recurrent neural networks. *Guide to OCR for Arabic Scripts*, 1–8. <http://doi.org/10.1007/978-1-4471-4072-6>
- He, X., Bowers, S., Candela, J. Q., Pan, J., Jin, O., Xu, T., ... Herbrich, R. (2014). Practical Lessons from Predicting Clicks on Ads at Facebook. *Proceedings of 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining - ADKDD'14*, 1–9. <http://doi.org/10.1145/2648584.2648589>
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory, 9(8), 1–32.
- Java, A., Song, X., Finin, T., & Tseng, B. (2007). Why We Twitter: Understanding Microblogging. *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis*, 56–65. <http://doi.org/10.1145/1348549.1348556>

- Krizhevsky, a., Sutskever, I., & Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 1097–1105.
- Lawrence, S., Giles, C. L., Tsoi, A. C., & Back, A. D. (1997). Face Recognition : A Convolutional Neural Network Approach. *Neural Networks*, 8(1), 98–113.
- Maass, W., Natschläger, T., Markram, H., Maass, W., Natschlaeger, T., Markram, H., & Maass, W. (2001). No Title, 1–27.
- Macskassy, S. a, & Michelson, M. (2011). Why Do People Retweet ? Anti-Homophily Wins the Day ! *Artificial Intelligence*, 209–216.
<http://doi.org/10.1144/GSL.SP.1999.156.01.07>
- McCulloch, W., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4), 115–133.
<http://doi.org/10.1007/BF02478259>
- McMahan, H. B., Holt, G., Sculley, D., Young, M., Ebner, D., Grady, J., ... Kubica, J. (2013). Ad click prediction: a view from the trenches. *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1222–1230. <http://doi.org/10.1145/2487575.2488200>
- Metaxas, P. T., Mustafaraj, E., Wong, K., Zeng, L., Keefe, M. O., & Finn, S. (2013). Do Retweets indicate Interest , Trust , Agreement ? (Extended Abstract).
- Monner, D., & Reggia, J. a. (2012). A generalized LSTM-like training algorithm for second-order recurrent neural networks. *Neural Networks*, 25(301), 70–83.
<http://doi.org/10.1016/j.neunet.2011.07.003>
- Peng, H.-K., Zhu, J., Piao, D., Yan, R., & Zhang, Y. (2011). Retweet Modeling Using Conditional Random Fields. *2011 IEEE 11th International Conference on Data Mining Workshops*, 336–343. <http://doi.org/10.1109/ICDMW.2011.146>
- Petrovic, S., Osborne, M., & Lavrenko, V. (2011). Rt to win! predicting message propagation in twitter. *Prof. of AAAI on Weblogs and Social ...*, 13, 586–589.

Retrieved from
<http://homepages.inf.ed.ac.uk/miles/papers/icwsm11.pdf>
<http://www.aaai.org/ojs/index.php/ICWSM/ICWSM11/paper/viewPDFInterstitial/2754/3209>

Sanzgiri, A. a, & Asnani, K. (2015). S Tock I Ndex P Rediction U Sing N Eural, 1–7.

Suh, B., Hong, L., Pirolli, P., & Chi, E. H. (2010). Want to be retweeted? Large scale analytics on factors impacting retweet in twitter network. *Proceedings - SocialCom 2010: 2nd IEEE International Conference on Social Computing, PASSAT 2010: 2nd IEEE International Conference on Privacy, Security, Risk and Trust*, 177–184. <http://doi.org/10.1109/SocialCom.2010.33>

Vranica, S. (2014). Behind the Preplanned Oscar Selfie: Samsung's Ad Strategy.

Retrieved April 9, 2015, from
<http://www.wsj.com/articles/SB10001424052702304585004579417533278962674>

Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1-3), 37–52. [http://doi.org/10.1016/0169-7439\(87\)80084-9](http://doi.org/10.1016/0169-7439(87)80084-9)

Wu, M. H. (2015). Financial Market Prediction.

Zaman, T., Fox, E. B., & Bradlow, E. T. (2013). A Bayesian Approach for Predicting the Popularity of Tweets. *arXiv Preprint*, 8(3), 28. <http://doi.org/10.1214/14-AOAS741>

Zaman, T. R., Herbrich, R., & Stern, D. (2010). Predicting Information Spreading in Twitter. *Social Science and*, 55(114171), 1–4. Retrieved from http://research.microsoft.com/pubs/141866/NIPS10_Twitter_final.pdf