UNIVERSIDAD NACIONAL DEL ALTIPLANO
REVISTA DE INVESTIGACIONES DE LA ESCUELA DE POSGRADO
Versión electrónica ISSN 2077–8686
Versión Impresa ISSN 1997-4035
ARTÍCULO ORIGINAL

Vol. 9 No.2, pp. 1530-1544, abril/junio 2020
Recibido 05/02/2020
Aceptado 27/04/2020
Publicado 30/04/2020
DOI: http://dx.doi.org/10.26788/riepg.2020.2.169

ANÁLISIS DE OPINIÓN EN TWITTER POR LA CLASIFICACIÓN AL MUNDIAL RUSIA 2018 DE LA SELECCIÓN PERUANA DE FÚTBOL CON SPARK

ANALYSIS OF OPINION ON TWITTER FOR THE 2018 RUSSIAN WORLD CLASSIFICATION OF THE PERUVIAN FOOTBALL TEAM WITH SPARK

Mayenka Fernández Chambi[1]

[1]Universidad Nacional del Altiplano Puno, Escuela Profesional de Ingeniería de Sistemas, Av. Floral N°1153, Puno, Perú, mayenkaf@gmail.com

RESUMEN

La presente investigación muestra el análisis de opinión realizado en los tuits históricos publicados en la red social o microblogging Twitter en idioma español durante el evento clasificatorio de la selección peruana de futbol al mundial Rusia 2018, durante el periodo del año 2015 hasta diciembre del 2017 según calendario clasificatorio Rusia 2018 de la FIFA. El modelo del análisis de opinión o sentimiento ha sido desarrollado en la plataforma de computación distribuida Spark; demostrándose que las tareas de preparación de datos, modelado y evaluación de algoritmos de aprendizaje de máquina para clasificación de texto se han desarrollado con eficiencia dentro del pipeline de Spark entre tareas transformadoras y estimadoras sobre la estructura de datos DataFrame y la librería MLlib, así los modelos estándar de aprendizaje de máquina para Big Data pueden ser realizadas en forma escalable y distribuida con facilidad de uso por los científicos de datos. Finalmente el modelo de clasificación binario de texto de tuits ha alcanzado una precisión de 83,51% para un modelo de regresión logística y está sobre las métricas estándar de aceptación de clasificadores de su mismo tipo; adicionalmente, esta investigación deja construido y disponible el dataset "PeruARusia2018.csv" con 3 000 ítems de tuits etiquetados siguiendo los estándares adecuados que la hacen propicia para que la comunidad investigadora pueda seguir experimentando sobre ella y halle mejores resultados; así como 376 250 tuits como raw data.

Palabras Clave: Análisis de opinión, big data, clasificación de texto, MLlib, red social, Spark.

ABSTRACT

The present investigation shows the analysis of opinion carried out in the historical tweets published in the social network or microblogging Twitter in spanish language during the qualifying event of the Peruvian soccer team to the Russia-2018 World Cup, during the period of the year 2015 until December of 2017 according to FIFA 2018 Russia qualification calendar. The opinion or sentiment analysis has been developed on the Spark distributed computating platform; demostrating that the tasks of data preparation, modeling and evaluation of machine learning algorithms for text classification has been efficiently developed within the Spark pipeline between transforming and estimating tasks on the DataFrame data structure and the MLlib library; thus, standard machine learning models for Big Data can be scalep up and distributed with ease of use by data scientists. Finally, the binary text classification model of tweets has reached an accuracy of 83,51% for a logistic regression model and is on the standard acceptance metrics of classifieres of the same type; additionally, this research leaves the "PeruARusia2018.csv" dataset built and available with 3 000 ítems of tweets labeled following the appropiate standards that make it conducive for the research community to continue experimenting on it and find better results; as well as 376 250 tuits like raw data.

Keywords: Opinion analysis, big data, text classification, MLlib, social network, Spark.

*Autor para correspondencia: mayenkaf@gmail.com

UNIVERSIDAD NACIONAL DEL ALTIPLANO
REVISTA DE INVESTIGACIONES DE LA ESCUELA DE POSGRADO
Versión electrónica ISSN 2077–8686
Versión Impresa ISSN 1997-4035
ARTÍCULO ORIGINAL

Vol. 9 No.2, pp. 1530-1544, abril/junio 2020
Recibido 05/02/2020
Aceptado 27/04/2020
Publicado 30/04/2020
DOI: http://dx.doi.org/10.26788/riepg.2020.2.169

INTRODUCCIÓN

El uso creciente de las redes sociales, correo electrónico, mensajes instantáneos de texto, chats en tiempo real, y los tuits, han propiciado el nacimiento de una nueva disciplina dentro de la minería de texto llamada análisis de opinión o sentimiento, se estima que más del 80% de datos son no estructurados y tienen la forma de texto (Aguilar, 2016), donde las personas son consideradas como sensores subjetivos del mundo real, ya que lo perciben y expresan su opinión en forma de texto sobre lo que les gusto o disgustó masivamente a través de las redes sociales cuando publican sus opiniones de lo que acontece (Zhai & Massung, 2016). Twitter es una red social y un gran depósito masivo que acumula opiniones (Pang & Lee, 2008), lanzado en el año 2006 se convirtió en un fenómeno masivo ya que la plataforma registró más de 200 millones de usuarios activos en 33 idiomas diferentes en el 2013 en el que se publicaron más de 400 millones de tuits por día (Witten *et al.*, 2016). Para tratar de comprender la opinión del texto de los tuits a través de medios computacionales se aplican algoritmos o modelos de aprendizaje de máquina supervisados y no supervisados, hallándose marcas de eficiencia del modelo aplicado (Go *et al.*, 2009), y se han establecido las etapas del proceso de análisis de opinión sin incluir propiamente la construcción del dataset (Liu, 2012), ya que se ha ido usando el dataset Sentiment140 mayormente para probar los algoritmos, más no su construcción misma (McMinn *et al.*,2013). Estas investigaciones además han sido realizadas aplicando modelos de arquitectura centralizada basadas en herramientas de minería de datos como Weka, y ScikitLearn pero limitado al procesamiento de datasets pequeños (Nodarakis *et al.*, 2016); mientras que los datasets que sobrepasan el espacio de memoria o los considerados Big Data necesitan de herramientas de minería de datos de arquitectura distribuida (Baltas *et al.*, 2016); la migración de modelos de análisis de opinión centralizados a modelos con arquitectura distribuida son necesarios y pueden ser realizados con herramientas como Spark que está diseñado para procesar grandes volúmenes de datos (Svyatkovskiy *et al.*, 2016), la masividad de los datos se producen en redes sociales como Twitter y sobre todo cuando acontecen eventos mundiales que estimulan la publicación masiva de tuits (McMinn *et al.*, 2013), tal como es el caso de las copas mundiales de futbol (FIFA.com, 2018) por lo que existirán grandes volúmenes de datos disponibles para estas aproximaciones.

Diversos estudios se han realizado sobre el análisis de opinión en Twitter con Spark. Un estudio probó que el modelo de análisis de opinión usando algoritmos de aprendizaje supervisado de MLlib de clasificación binaria y multiclase kNN de Spark 1.4.1 junto a filtros Bloom fue eficiente, robusto y linealmente escalable; el dataset utilizado fue construido arrastrando 942 188 tuits con hashtags y 1 337 508 tuits con emoticones entre noviembre de 2014 hasta agosto de 2015, el cual fue etiquetado explotando los hashtags y emoticones validados por jueces humanos, y cuyo modelo fue evaluado usando el método de validación cruzada de 10 dobles basado en su exactitud (Nodarakis *et al.*, 2016). Otro estudio implementó un sistema de análisis de opinión con MLlib de Spark para medir el efecto que causa el tamaño y número de características del dataset de entrenamiento en la mejora de la exactitud del clasificador binario y ternario basado en modelos de Redes Bayesianas, Regresión Logística y Árboles de decisión al introducir procesamiento de lenguaje natural en el pre-procesamiento (Baltas *et al.*, 2016). Otro estudio más reciente muestra las marcas de desempeño alcanzados por modelos y tendencias tecnológicas del análisis de opinión de la quinta edición del SemEval realizado sobre datasets con tuits en inglés y árabe publicados entre setiembre de 2016 a enero de 2017, el análisis de opinión fue realizado usando modelos de Entropía Máxima, Regresión Logística, Redes Bayesianas, SVM y herramientas como Sklearn, Numpy, Python, Java, TensorFlow, Weka, Theano, y Stanford CoreNLP (Rosenthal *et al.*, 2017). En cuanto a la construcción del dataset para análisis de opinión, el primer trabajo de análisis de opinión en Twitter estableció las clases de sentimiento que se pueden hallar dentro

UNIVERSIDAD NACIONAL DEL ALTIPLANO
REVISTA DE INVESTIGACIONES DE LA ESCUELA DE POSGRADO
Versión electrónica ISSN 2077–8686
Versión Impresa ISSN 1997-4035
ARTÍCULO ORIGINAL

Vol. 9 No.2, pp. 1530-1544, abril/junio 2020
Recibido 05/02/2020
Aceptado 27/04/2020
Publicado 30/04/2020
DOI: http://dx.doi.org/10.26788/riepg.2020.2.169

del texto de un tuit: positivo, negativo y neutro; a su vez mostró la construcción del dataset Sentiment140 para estos propósitos (Go *et al.*, 2009).

El análisis de opinión es una especialización de la minería de texto, la minería de texto se relaciona con la minería web y ambas son descendientes de la minería de datos (Liu, 2011); la minería de datos es el proceso de automáticamente descubrir información útil y enfrenta desafíos de escalabilidad, alta dimensión en las características, y heterogeneidad de los datos (Tan *et al.*, 2006), como disciplina provee también herramientas para procesar datos generados en las redes sociales (Zafarani *et al.*, 2014); estas herramientas o modelos son de tipo estadístico, aprendizaje de máquina, resúmenes, y de extracción de características (Leskovec *et al.*, 2014).

El proceso de minería de texto en Big Data está compuesto en recuperar información sin procesamiento en uno más pequeño pero relevante, y luego aplicar modelos de minería de texto que permita descubrir el conocimiento (Zhai & Massung, 2016), la minería de texto que analiza las opiniones de las personas sobre productos, servicios, organizaciones, individuos, temas, y eventos es importantes en toda actividad humana, ya que la toma de decisiones está basada e influenciada por las opiniones de otros, por ejemplo un comprador busca conocer la opinión de sus pares con respecto al producto que pretende comprar, o una empresa espera conocer las opiniones de sus clientes con respecto a la aceptación o rechazo de sus productos y/o servicios (Pang & Lee, 2008). Una opinión es una declaración subjetiva que describe lo que una persona cree o piensa sobre algo, la subjetividad es un factor diferenciador importante que mide si es positiva o negativa con respecto al contenido de la opinión y no si es correcta o incorrecta en declaraciones objetivas. Por lo tanto, el análisis de opinión recibe como entrada un objeto de texto opinado y produce una salida que es una etiqueta de opinión o sentimiento categorizada como positiva o negativa (Zhai & Massung, 2016).

Los modelos de aprendizaje de máquina que se aplican en el análisis de opinión son de aprendizaje supervisado, donde los valores de la etiqueta del dataset son conocidos para entrenar el modelo (Zafarani *et al.*, 2014); la regresión logística es un caso especial de los modelos lineales y es un modelo de clasificación de texto adecuado para el análisis de opinión ya que su objetivo es predecir una respuesta binaria al medir probabilísticamente la relación entre la etiqueta “Y” y sus características “X” usando una función logística y normalizándola con una función sigmoide (Singh, 2018). La evaluación de los modelos de análisis de opinión se realiza calculando la exactitud, precisión, y recuperación alcanzada en la fase de prueba; la exactitud es la suma de los positivos y negativos verdaderos dividido entre el número total de registros del dataset, la precisión es el número actual de casos positivos fuera de los casos positivos pronosticados por el modelo, y la recuperación es el porcentaje actual de los casos positivos que el modelo se permite pronosticar correctamente fuera del número total de casos positivos (Singh, 2018).

Spark es un sistema computacional distribuido unificado más un conjunto de librerías que permiten el procesamiento en paralelo a alta velocidad de grandes cantidades de datos en forma escalable y rápida (Zaharia *et al.*, 2016); provee de herramienta estándar para procesar Big Data programados en Python, Java, Scala y R, proporciona también una API MLlib y ML estándar que permite realizar procesos de aprendizaje de máquina en cada una de ellas como RegressionLogistic, los datos se organizan fácil y eficientemente con DataFrames con forma de tabla más esquema y pueden ser utilizados por las APIs de transformación que modifican los datos en forma inmutable y las de acción producen las ejecuciones (Chambers & Zaharia, 2018). Finalmente, Spark ML introducido en la versión 2.0 permite el pre procesamiento, transformación de datos, entrenamiento de modelos, y realización de predicciones usando DataFrames (AI Zone, 2019) junto a APIs adicionales para tareas de extracción, transformación

UNIVERSIDAD NACIONAL DEL ALTIPLANO
REVISTA DE INVESTIGACIONES DE LA ESCUELA DE POSGRADO
Versión electrónica ISSN 2077–8686
Versión Impresa ISSN 1997-4035
ARTÍCULO ORIGINAL

Vol. 9 No.2, pp. 1530-1544, abril/junio 2020
Recibido 05/02/2020
Aceptado 27/04/2020
Publicado 30/04/2020
DOI: http://dx.doi.org/10.26788/riepg.2020.2.169

y selección de características como HashingTF, StopWordsRemover, y Tokenizer (Apache Spark, 2019).

Por lo tanto, esta investigación trató de realizar el análisis de opinión de tuits publicados por la clasificación al mundial Rusia 2018 en una arquitectura distribuida como Spark para medir su eficiencia con respecto a analizadores de arquitectura centralizada, para ello además, se planteó dos objetivos específicos: primero, construir el dataset de tuits por la clasificación al mundial de futbol Rusia 2018 de la selección peruana de futbol y segundo, pre procesar el dataset, entrenar y evaluar el modelo de análisis de opinión usando Spark para medir su eficacia. Así mismo se plantearon las siguientes hipótesis específicas: primero, el dataset de tuits por la clasificación al mundial de futbol Rusia 2018 de la selección peruana de futbol tiene las mismas características estándar del dataset Sentiment140 de análisis de opinión en Twitter; y segundo, la exactitud del modelo de análisis de opinión en Twitter por la clasificación al mundial de futbol Rusia 2018 de la selección peruana de futbol con Spark es mayor a la exactitud promedio de los modelos de SemEval-2017 Task 4: Message Polarity Classification.

MATERIALES Y MÉTODOS

Lugar de estudio

La investigación fue experimental tecnológica de dos variables, la variable independiente fue el dataset conformado por tuits históricos publicados sobre la clasificación al mundial de futbol Rusia 2018 de la selección peruana de futbol en idioma español y se midió sus efectos en la variable dependiente que fue el modelo de análisis de opinión. Esta investigación se dividió en dos etapas conforme a los objetivos específicos planteados, primero en construir el dataset para aprendizaje supervisado con los tuits históricos de la selección peruana de futbol, y la segunda en modelar el analizador de opinión sobre el dataset construido usando Spark; ambas etapas fueron realizadas en la Universidad Nacional del Altiplano de Puno.

La población se supuso desconocida y muy grande, ya que se ignoraba cuántos tuits exactamente se publicaron desde octubre de 2015 hasta noviembre de 2017 con relación a la clasificación al mundial de Rusia 2018 de la selección peruana de futbol, por lo que, la elección de la muestra fue de tipo no probabilístico y se utilizó el muestreo casual o incidental, eligiendo los tuits históricos que estuvieron accesibles al arrastre de tuits y que puedan ser etiquetadas manualmente.

Descripción detallada por objetivos específicos

Construir el dataset de tuits para el análisis de opinión en Twitter por la clasificación al mundial de futbol Rusia 2018 de la selección peruana de futbol.

Para construir legalmente un cuerpo de datos de Twitter con el mismo esquema del dataset Sentiment140 se siguió dos fases:

La primera fase de arrastre de tuits históricos se aplicó el método de un arrastrador HTTP, que recupera tuits usando asíncrono para obtener tuits individualmente desde el sitio de Twitter.com y reconstruirlos en formato CSV sin usar directamente la API Rest de Twitter (McCreadie *et al.*, 2012) (Figura 1).

UNIVERSIDAD NACIONAL DEL ALTIPLANO
REVISTA DE INVESTIGACIONES DE LA ESCUELA DE POSGRADO
Versión electrónica ISSN 2077–8686
Versión Impresa ISSN 1997-4035
ARTÍCULO ORIGINAL

Vol. 9 No.2, pp. 1530-1544, abril/junio 2020
Recibido 05/02/2020
Aceptado 27/04/2020
Publicado 30/04/2020
DOI: http://dx.doi.org/10.26788/riepg.2020.2.169

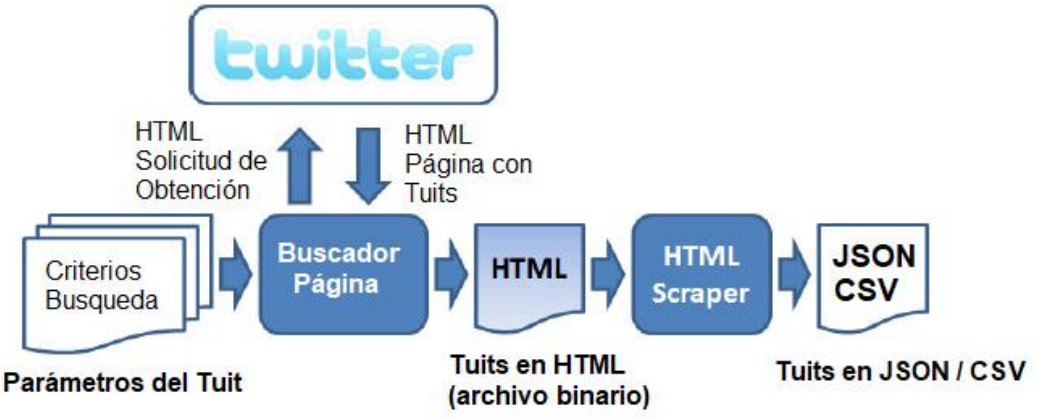

Figura 1. Arrastrador HTTP Asíncrono de tuits históricos.

El método estadístico para evaluar la hipótesis específica 1: el dataset de tuits por la clasificación al mundial de futbol Rusia 2018 de la selección peruana de futbol tiene las mismas características estándar del dataset Sentiment140 de análisis de opinión en Twitter, se utilizó la prueba de hipótesis sobre la media t, en función al número de características que alcance el dataset construido con respecto al dataset Sentimento140 (Go *et al.*, 2009).

$$t = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}} \; (n-1)GL$$

La segunda fase se aplicó el método de etiquetado manual de cada tuit a través de una aplicación web y producir el dataset final (McMinn *et al.*, 2013) (Figura 2).

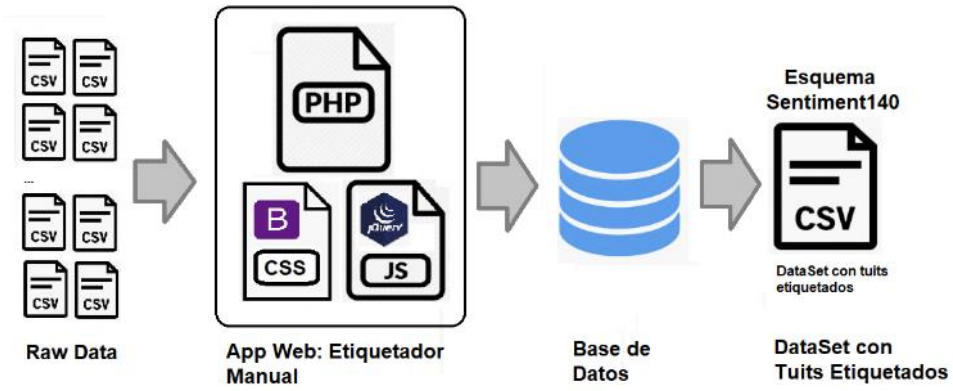

Figura 2. Aplicación Web para el etiquetado manual de tuits.

En cuanto a los materiales, se usó la API de código libre GetOldTweets-python (Jefferson, 2016) que arrastra tuits históricos pasados y el lenguaje de programación Python 3.4; mientras que para construir el sistema de etiquetado manual se usó un ambiente de programación Web basado en: PHP, JQuery, y Bootstrap 4; junto a la base de datos MySQL 5.7.

UNIVERSIDAD NACIONAL DEL ALTIPLANO
REVISTA DE INVESTIGACIONES DE LA ESCUELA DE POSGRADO
Versión electrónica ISSN 2077–8686
Versión Impresa ISSN 1997-4035
ARTÍCULO ORIGINAL

Vol. 9 No.2, pp. 1530-1544, abril/junio 2020
Recibido 05/02/2020
Aceptado 27/04/2020
Publicado 30/04/2020
DOI: http://dx.doi.org/10.26788/riepg.2020.2.169

Pre-procesar el dataset, entrenar y evaluar el modelo de análisis de opinión usando Spark para analizar las opiniones en Twitter por la clasificación al mundial Rusia 2018 de la selección peruana de fútbol.

Para programar el modelo de análisis de opinión se siguió la metodología del proceso de análisis de opinión el cual consistente de seis etapas: cargar el dataset de tuits, pre procesar el texto que limpie el texto, tokenizar el texto pre procesado, aplicar ingeniería de características que elimine palabras sin utilidad del idioma y las vectorice numéricamente, construir el modelo de clasificación y finalmente evaluar la eficiencia del modelo (Liu, 2012) (Figura 3).

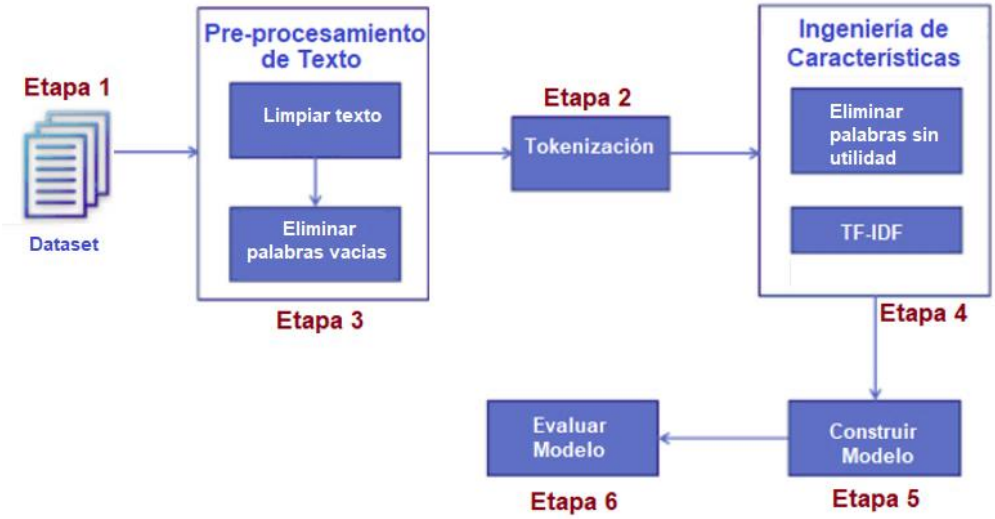

Figura 3. Proceso de análisis de opinión.

El método estadístico para evaluar la hipótesis específica 2: la exactitud del modelo de análisis de opinión en Twitter por la clasificación al mundial de futbol Rusia 2018 de la selección peruana de futbol con Spark es mayor a la exactitud promedio de los modelos de SemEval-2017 Task 4: Message Polarity Classification, se utilizó la prueba de hipótesis sobre la media z, en función de la exactitud que obtuvo el modelo con respecto al promedio alcanzado de los modelos utilizados en SemEval (Rosenthal, Farra, y Nakov, 2017).

$$Z = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}} \sim N(0,1)$$

La valoración numérica de la eficiencia del modelo se realizó aplicando las fórmulas compuestas por la exactitud, precisión y recuperación del modelo entrenado con un dataset de entrenamiento con respecto a un dataset de prueba (Singh, 2018) (Tabla 1).

Tabla 1. Fórmulas de medidas de evaluación del modelo de análisis de opinión.

Medida	Formula
Exactitud (Accuracy)	$Exactitud = \frac{VP + VN}{VP + VN + FP + FN}$
Precisión (Precision)	$Precisión = \frac{VP}{VP + FP}$
Recuperación / Sensibilidad (Recall)	$Recuperación\ o\ Sensibilidad = \frac{VP}{VP + FN}$

Fuente: Adaptado de (Singh, 2018)

UNIVERSIDAD NACIONAL DEL ALTIPLANO
REVISTA DE INVESTIGACIONES DE LA ESCUELA DE POSGRADO
Versión electrónica ISSN 2077–8686
Versión Impresa ISSN 1997-4035
ARTÍCULO ORIGINAL

Vol. 9 No.2, pp. 1530-1544, abril/junio 2020
Recibido 05/02/2020
Aceptado 27/04/2020
Publicado 30/04/2020
DOI: http://dx.doi.org/10.26788/riepg.2020.2.169

En cuanto a los materiales, se programó en Spark 2.3.1 con Hadoop 2.7 en modo local en un sistema operativo Windows 10 Pro (I7-5600U CPU, 8Gb RAM); la programación se realizó en el Notebook interactivo Jupyter-Python a través del servidor Anaconda 3 para Python 3.x, las APIs aplicadas son las que proporciona el Spark basadas en: SparkSession, DataFrame y los algoritmos de aprendizaje de máquina MLlib de PySpark: Tokenizer, Regex, StopWordsRemover, LogisticRegression, y HashingTF.

RESULTADOS Y DISCUSIÓN

Construir el dataset de tuits para el análisis de opinión en Twitter por la clasificación al mundial de futbol Rusia 2018 de la selección peruana de futbol.

Se arrastró tuits históricos usando una aplicación programada en Python 3.4 y la API GetOldTweets-python (Jefferson, 2016) que cumplieron con parámetros de búsqueda de intervalos de fecha combinadas con palabras hashtags y nombres de cuenta. Además se usaron 11 intervalos de fechas de búsqueda correspondientes al calendario clasificatorio de la FIFA Ronda 1 y Play-Off Copa Rusia 2018 (FIFA.com, 2018).

A continuación se muestran 162 hashtags de la selección peruana obtenidos por exploración manual en Twitter.com junto a la aplicación web de tendencias (Trendogate.com, 2018) (Tabla 2).

Tabla 2. Algunos intervalos de fechas de búsqueda para arrastrar tuits de la selección peruana de futbol.

Nro	Intervalo de fecha búsqueda	Fecha partido de fútbol FPF	Días
1	Del 07 Oct 2015 al 15 Oct 2015	09 Oct 2015 y 13 Oct 2015	09
2	Del 11 Nov 2015 al 19 Nov 2015	13 Nov 2015 y17 Nov 2015	09
3	Del 12 Mar 2016 al 16 Mar 2016	14 Mar 2016	05
4	Del 29 Ago 2017 al 07 Set 2017	31 Ago 2017 y 05 Set 2017	10
5	Del 09 Nov 2017 al 17 Nov 2017	11 Nov 2017 y15 Nov 2017	09

Fuente: Obtenido de (FIFA.com, 2018).

Seguidamente se muestran algunos hashtags en Twitter recolectados de cuentas oficiales de la Federación Peruana de Futbol, clubes, periodistas, programas, periódicos deportivos (Tabla 3).

Tabla 3. Algunos hashtags de búsqueda para arrastrar tuits de la selección peruana de futbol.

Nro	Match	Fecha	HashTags Trending Topic
1	Colombia – Perú	09 Oct 2015	#PerdimosComoSiempre
2	Paraguay – Perú	10 Nov 2016	#ArribaPeru, #Rusia2018
3	Venezuela – Perú	23 Mar 2017	#CHONGOPERU4ANO
4	Ecuador – Perú	05 Set 2017	#SiPerúGanaPrometo
5	Argentina – Perú	05 Oct 2017	#selecciónperuana

Fuente: Obtenido de (Trendogate.com, 2018).

Se identificaron 80 nombres de cuentas de usuario más jugadores de la selección peruana; se muestran algunos ejemplos (Tabla 4).

UNIVERSIDAD NACIONAL DEL ALTIPLANO
REVISTA DE INVESTIGACIONES DE LA ESCUELA DE POSGRADO
Versión electrónica ISSN 2077–8686
Versión Impresa ISSN 1997-4035
ARTÍCULO ORIGINAL

Vol. 9 No.2, pp. 1530-1544, abril/junio 2020
Recibido 05/02/2020
Aceptado 27/04/2020
Publicado 30/04/2020
DOI: http://dx.doi.org/10.26788/riepg.2020.2.169

Tabla 4. Algunas cuentas de usuario de búsqueda para arrastrar tuits de la selección peruana de futbol.

N°	Nombre	Cuenta	Seguidores	Categoría
1	Marca	@marca	5.2 M	Portal deportivo
2	CONMEBOL.COM	@CONMEBOL	1.2 M	Confederación
3	Federación Peruana de Futbol	@TuFPF	1.1 M	Federación
4	Club Universitario de Deportes	@Universitario	863.6 K	Club deportivo
5	Eddie Fleishman	@E_FLEISCHMAN	745.5 K	Periodista deportivo

La aplicación de arrastre de tuits se compuso de dos métodos: descargar y orquestar, el primero arrastra N tuits y genera un archivo csv, y el segundo, orquesta la repetición de arrastre para todas las combinaciones buscadas entre fechas y hashtags o fechas y cuentas. Se obtuvo finalmente 2 664 archivos csv, 1 784 por intervalos de fecha y hashtags, y 880 por intervalos de fecha y nombre de cuenta, cada archivo contuvo 500 tuits en caso que el arrastre fue totalmente exitoso, entre 1 y 499 tuits en caso de éxito medio y 0 tuits en caso de fracaso. Así, el número de tuits arrastrado en promedio fue de 376 250 lo que constituyó el raw data total obtenido; 216 750 por intervalos de fecha y hashtags y 159 500 por intervalos de fecha y nombre (Tabla 5).

Tabla 5. Número total de tuits arrastrados y recuperados exitosamente en archivos csv.

Criterio	Categoría de Éxito	Archivos CSV	Tuits Arrastrados
Intervalo de Fecha de búsqueda y Hashtags	500 Tuits	249	124 500
	Menos de 500 Tuits	369	92 250
	0 Tuits	1 166	0
	Sub Total	**1 784**	**216 750**
Intervalo de Fecha de búsqueda y Cuentas	500 Tuits	67	33 500
	Menos de 500 Tuits	504	126 000
	0 Tuits	309	0
	Sub Total	**880**	**159 500**
	Total	**2 664**	**376 250**

Para completar la construcción del dataset se programó una aplicación web que consumió archivos csv individuales del raw data total, y desde la interfaz de usuario se etiquetó manualmente cada tuit recuperado, centralizando los tuits etiquetados en una base de datos en MySQL; desde el cual se generó el dataset resultante "PeruARusia2018.csv" con el mismo esquema del dataset Sentiment140; la interfaz de usuario de la aplicación web (Figura 4).

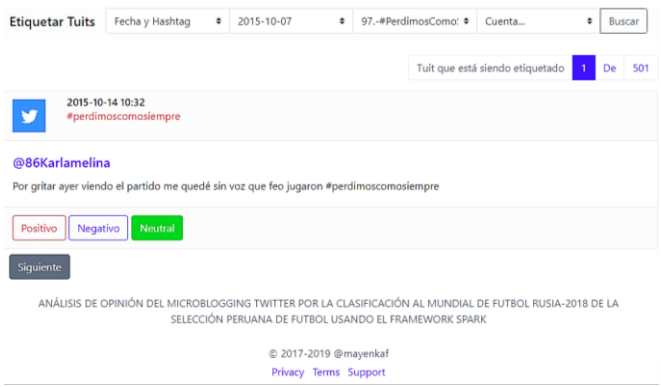

Figura 4. Interfaz de la aplicación web para el etiquetado manual de tuits

UNIVERSIDAD NACIONAL DEL ALTIPLANO
REVISTA DE INVESTIGACIONES DE LA ESCUELA DE POSGRADO
Versión electrónica ISSN 2077–8686
Versión Impresa ISSN 1997-4035
ARTÍCULO ORIGINAL

Vol. 9 No.2, pp. 1530-1544, abril/junio 2020
Recibido 05/02/2020
Aceptado 27/04/2020
Publicado 30/04/2020
DOI: http://dx.doi.org/10.26788/riepg.2020.2.169

Finalmente se etiquetó 5 000 tuits manualmente, del que se obtuvo 1 500 tuits con etiqueta negativa y 3 500 con etiqueta positiva, a partir del cual se obtuvo el dataset final en formato CSV simétrico de 3 000 tuits, 1 500 tuits negativos y 1 500 tuits positivos llamado PeruARuisa2018.csv**.**

Se construyó el dataset PeruARusia2018.csv que reunió 9 características de las 12 características de un dataset legalmente construido como el dataset Sentiment140 (Go *et al.*, 2009), cuyas características cumplidas y no cumplidas (Tabla 6).

Tabla 6. Características cubiertas por el dataset PeruARusia2018 con respecto al dataset Sentiment140.

12 características de un dataset Legalmente construido	Medida Cuantitativa Sentiment140	Medida Cuantitativa PeruARusia2018
Archivo externo CSV	Si = 1	Si = 1
Formato de Texto UTF-8	Si = 1	Si = 1
Balanceado en 2 clases (Tuits positivos es igual a Tuits negativos)	No = 0	Si = 1
Balanceado en 3 clases (Hay tantos tuis positivos como negativos y nulos)	Si = 1	No = 0
Esquema contiene: Id, Fecha Tuit, Consulta, Usuario que publicó el tuit, Texto intacto del Tuit, Etiqueta del Tuit	Si = 6	Si = 6
Nro de Tuits = 1600000	Si = 1	No = 0
El texto del tuit contiene emoticonos	No = 0	No = 0
Total	**10 de 12**	**9 de 12**

Se realizó la prueba de la hipótesis específica 1**,** donde H_0: El dataset para el análisis de opinión en Twitter por la clasificación al mundial de futbol Rusia 2018 de la selección peruana de futbol no tiene las mismas características estándar del dataset Sentiment140 (H_0: $\bar{X}_{dsPeruARusia} \neq \mu_{dsSentiment140}$) y H_1: El dataset para el análisis de opinión en Twitter por la clasificación al mundial de futbol Rusia 2018 de la selección peruana de futbol tiene las mismas características estándar del dataset Sentiment140 (H_1: $\bar{X}_{dsPeruARusia} = \mu_{dsSentiment140}$), con un nivel de significancia de 0,02 o 2% de error o $\alpha = 0{,}02 = 2\%$ y $GL = 11$, se utilizó la distribución t, obteniéndose $t_\alpha = t_{0{,}02} = -2{,}718\ y\ 2{,}718$. Se usó el estadístico de prueba para $n = 12$; n < 30; donde $n = 12$ es el número de características de un dataset.

Estadística de Prueba: Se ha usado los datos proporcionados por la tabla 6 en el remplazo de la ecuación.

$$t_1 = \frac{9 - 10}{0.577/\sqrt{12}} = -5{,}477$$

Luego de realizada la prueba estadística, rechazamos la Hipótesis nula H_0 por lo tanto se acepta la hipótesis alterna H_1, así que el dataset para el análisis de opinión del microblogging Twitter por la clasificación al mundial de futbol Rusia-2018 de la selección peruana de futbol tiene las mismas características estándar del dataset Sentiment140, por lo que es adecuado para utilizarla en modelos de clasificación de texto.

El método de obtención de tuits históricos usando un extractor HTTP asíncrono (McCreadie *et al.*, 2012) funcionó exitosamente ya que se logró arrastrar 376 250 tuits como raw data con la ejecución del programa arrastrador limitado a sólo 500 tuits por combinación de búsqueda de la selección peruana de

UNIVERSIDAD NACIONAL DEL ALTIPLANO
REVISTA DE INVESTIGACIONES DE LA ESCUELA DE POSGRADO
Versión electrónica ISSN 2077–8686
Versión Impresa ISSN 1997-4035
ARTÍCULO ORIGINAL

Vol. 9 No.2, pp. 1530-1544, abril/junio 2020
Recibido 05/02/2020
Aceptado 27/04/2020
Publicado 30/04/2020
DOI: http://dx.doi.org/10.26788/riepg.2020.2.169

futbol, siendo este número proporcional al que obtuvieron arrastrando todos los tuits con hashtags sin restricción de un evento de 942 188 entre noviembre de 2014 y agosto 2015 usando la API Rest de Twitter (Nodarakis *et al.*, 2016). Además este método permitió superar las restricciones de acceso a tuits históricos y límite del número de tuits por petición que impone el servicio REST de Twitter (Twitter.com, 2018), ya que el arrastre se produjo como si un usuario estuviera usando la función arrastre horizontal de la propia página web de Twitter y visualizando los tuits históricos (Jefferson, 2016).

El método de etiquetado manual usando una aplicación web (McMinn *et al.*, 2013) funciono regularmente ya que no se logró procesar todos los tuits del raw data, sino se limitó al esfuerzo de los etiquetadores humanos que tuvieron que leer cada tuit y dar su veredicto al etiquetarlo, sólo se logró etiquetar 5 000 tuits, de los que 3 500 fueron positivos y 1 500 negativos (muchos otros tuits leídos al no contener opinión fueron descartados y no contabilizados); que a diferencia del método aplicado por (Nodarakis *et al.*, 2016) y (Baltas *et al.*, 2016) si llegaron a cubrir todo el raw data arrastrado porque no usaron la etiquetación manual sino se supuso una etiqueta a aquellos tuits que contenían ciertos hashtags como etiquetas positivas o negativas validados sí por jueces humanos pero automáticamente. Aun así, se constató que al aplicar el etiquetado manual se pudo descartar aquellos tuits de tipo spam, retuits, y neutros o sin subjetivismo anticipadamente ya que no llegaron a formar parte del dataset final como si lo tuvieron el dataset de (Nodarakis *et al.*, 2016). Esta limitación redujo significativamente la cantidad de tuits contenidos en el dataset resultante PeruARusia 2018 a sólo 3 000 tuits ya que se pretendió que exista simetría en la cantidad de tuits etiquetados como positivos y negativos, cuyo límite se basó a sólo los 1 500 tuits etiquetados como negativos a través de la etiquetación manual.

Pre-procesar el dataset, entrenar y evaluar el modelo de análisis de opinión usando Spark para analizar las opiniones en Twitter por la clasificación al mundial Rusia 2018 de la selección peruana de fútbol.

Se utilizó el Spark 2.3.1 con Hadoop 2.7 en modo local en un sistema operativo Windows 10 Pro (I7-5600U CPU, 8Gb RAM); la programación se realizó en el Notebook interactivo Jupyter-Python a través del servidor Anaconda 3 para Python 3.x y Spark, se programó las 6 etapas del proceso de análisis de opinión de (Liu, 2012). En la etapa 1, se cargó y leyó el dataset PeruARusia2018.csv en el DataFrame "tuits_peruArusia" infiriendo automáticamente su esquema, luego se creó el DataFrame "tuitsData" con las columnas de entrada útiles para el modelo de análisis de opinión, estas incluyeron la columna que contenía la etiquetada (convertida de String a Int utilizando una función cast) y la columna que contenía el texto del tuit (Figura 5).

```
#importar librerias pyspark
from pyspark.sql.types import *
from pyspark.sql.functions import *
from pyspark.ml.classification import LogisticRegression
from pyspark.ml.feature import HashingTF, Tokenizer, StopWordsRemover
#crear sesion spark
spark = SparkSession.builder.appName("Analisis_Opinion_PeruARusia").config("spark.some.config.option","some-value")\
        .getOrCreate()
#cargar dataset en DataFrame
tuits_peruArusia = spark.read.csv("data/PeruARusia2018.csv", inferSchema = True, header = True)
#seleccionar los datos necesarios para el modelo de aprendizaje
tuitsData = tuits_peruArusia.select(col("polaridad").cast("Int").alias("label"),"texto")
```

Figura 5. Etapa 1 del análisis de opinión, cargar el dataset.

En la etapa 2, se programó y ejecutó 7 tareas de pre procesamiento en la columna "texto" para obtener un texto limpio, incluyó: remplazar las vocales atildadas "áéíóú" por vocales sin tilde "aeiou" utilizando la función translate, eliminar palabras que representan usuarios, hashtags, URLs y signos de puntuación, interrogación, admiración y el carácter espacio utilizando la función regexp_replace y patrones de

UNIVERSIDAD NACIONAL DEL ALTIPLANO
REVISTA DE INVESTIGACIONES DE LA ESCUELA DE POSGRADO
Versión electrónica ISSN 2077–8686
Versión Impresa ISSN 1997-4035
ARTÍCULO ORIGINAL

Vol. 9 No.2, pp. 1530-1544, abril/junio 2020
Recibido 05/02/2020
Aceptado 27/04/2020
Publicado 30/04/2020
DOI: http://dx.doi.org/10.26788/riepg.2020.2.169

expresión regular como "@[A-Za-z0-9]+|#[A-Za-z0-9]+", "http://[A-Za-z0-9]+/[A-Za-z0-9] | pic.twitter.com/[A-Za-z0-9]+ | goo.gl/[A-Za-z0-9]+", "\|\.|,|;|:|¿|\?|¡|!||=|-|…| + " por un carácter vacío "" (Figura 6).

```
#1.-limpiando texto, remplazando vocales atildadas por vocales simples
dataSinTilde = tuitsData.select(translate(col("texto"),"áéíóú","aeiou").alias("textoSinTilde")\
                                ,col("label"))
#2 y 3.-limpiando texto, quitando @usuarios y #etiqueta
patron = "@[A-Za-z0-9]+|#[A-Za-z0-9]+"
dataSinUserEtiqueta = dataSinTilde.select(regexp_replace(col("textoSinTilde"),patron,"").alias("textoSinUsEt")\
                                          ,col("label"))
#4 y 5-limpiandotexto,quitando pic.twitter.com/... goo.gl/...twitter.com/../.. /status/... fb.me/...
patron="pic.twitter.com/[A-Za-z0-9]+|goo.gl/[A-Za-z0-9]+|twitter.com/[A-Za-z0-9]+/[A-Za-z0-9]+\
|twitter.com/[A-Za-z0-9]+|/status/[A-Za-z0-9]+|fb.me/[A-Za-z0-9]+"
dataSinURL = dataSinUserEtiqueta.select(regexp_replace(col("textoSinUsEt"),patron,"").alias("textoSinURL")\
                                        ,col("label"))
#6.-limpiando texto, quitando simbolos, signos de puntuacion
patron="\"|\.|,|;|:|¿|\?|¡|!|=|-|/|…"
dataSinPunt = dataSinURLs.select(regexp_replace(col("textoSinURLs"),patron,"").alias("textoSinPunt")\
                                 ,col("label"))
#7.-limpiando texto, quitando espacios en blanco demas y remplazarlo por un espacio
patron=" +"
dataLimpio = dataSinPunt.select(regexp_replace(col("textoSinPunt"),patron," ").alias("text")\
                                ,col("label"))
```

Figura 6. Etapa 2 del análisis de opinión, pre procesamiento de texto.

Antes de la programación de las etapas 3, 4 y 5, se dividió el DataFrame resultado del pre procesamiento en dos partes: 70% formó parte de los datos de entrenamiento **tuitsEntrenamiento** y el 30% formó parte de los datos de prueba **tuitsPrueba**, cada ítem (tuit) de cada parte fue elegido usando la función randomSplit garantizando la aleatoriedad de ambas partes. La tokenización en palabras unigrama se realizó por espacios en blanco del texto, se quitó las palabras sin utilidad o StopWords usando la librería StopWordsRemover configurado para el idioma español que incluyó la eliminación de artículos como "la, el, las, los", adverbios, y tabs, se transformó las palabras del dataset de entrenamiento en vectores numéricos usando el transformador HashingTF**,** el mismo que calculó la frecuencia de las palabras presente en el texto del tuit, la etapa 5 incluyo el entrenamiento de un modelo de clasificación binaria basado en regresión logística con las siguientes características: entrada de entrenamiento fue la columna "features", y la de salida la columna "label", el modelo entrenó en 10 iteraciones con un parámetro de regularización de 0.01 (Figura 7).

```
#Dividir el dataset en entrenamiento=70% y prueba=30%
tuitsDividido = dataLimpio.randomSplit([0.7, 0.3])
tuitsEntrenamiento = tuitsDividido[0]
tuitsPrueba = tuitsDividido[1]
#Preparar Datos de Entrenamiento: Tokenizacion
tokenizer = Tokenizer(inputCol = "text", outputCol="palabras")
tokenizedData = tokenizer.transform(tuitsEntrenamiento)
#Preparar Datos: quitar palabras sin importancia de spanish
spanishStopWords = StopWordsRemover.loadDefaultStopWords("spanish")
swr = StopWordsRemover(stopWords = spanishStopWords, inputCol = tokenizer.getOutputCol(), outputCol="palabrasUtil")
swrTuitsData = swr.transform(tokenizedData)
#Convertir las palabras Utiles a caracteristicas numericas
hashTF = HashingTF(inputCol=swr.getOutputCol(),outputCol="features")
numericTrainData = hashTF.transform(swrTuitsData).select("label","palabrasUtil","features")
#Entrenar el modelo con los datos de entrenamiento
lr = LogisticRegression(labelCol="label",featuresCol="features",maxIter=10, regParam=0.01)
model = lr.fit(numericTrainData)
```

Figura 7. Etapas 3, 4 y 5 del análisis de opinión, tokenizar, ingeniería de características y entrenar el modelo.

UNIVERSIDAD NACIONAL DEL ALTIPLANO
REVISTA DE INVESTIGACIONES DE LA ESCUELA DE POSGRADO
Versión electrónica ISSN 2077–8686
Versión Impresa ISSN 1997-4035
ARTÍCULO ORIGINAL

Vol. 9 No.2, pp. 1530-1544, abril/junio 2020
Recibido 05/02/2020
Aceptado 27/04/2020
Publicado 30/04/2020
DOI: http://dx.doi.org/10.26788/riepg.2020.2.169

Finalmente en la etapa 6, se programó la evaluación del modelo calculando las medidas de evaluación del modelo de análisis de opinión: exactitud (*accuracy*), recuperación (*recall*), y precisión (*precision*) usando el modelo entrenado sobre el dataset de prueba (Figura 8).

```
#Predecir los datos de evaluación y calcular la eficiencia del modelo
prediction = model.transform(numericTest)
predictionFinal = prediction.select("palabrasUtil","prediction","label")
predictionFinal.show(n=5)
correctPrediction = predictionFinal.filter(predictionFinal["label"]==predictionFinal["prediction"]).count()
totalData = predictionFinal.count()
true_positives = predictionFinal.filter('label==1 and prediction==1.0').count()
true_negatives = predictionFinal.filter('label==0 and prediction==0.0').count()
false_positives = predictionFinal.filter('label==0 and prediction==1.0').count()
false_negatives = predictionFinal.filter('label==1 and prediction==0.0').count()
recall = float(true_positives)/(true_positives + false_negatives)
precision = float(true_positives)/(true_positives + false_positives)
accuracy = float(true_positives + true_negatives)/(totalData)
```

Figura 8. Etapa 6 del análisis de opinión, evaluación del modelo de análisis de opinión.

Se realizó la prueba de la hipótesis específica 2, donde H_0: la exactitud del modelo de análisis de opinión en Twitter por la clasificación al mundial de futbol Rusia 2018 de la selección peruana de futbol con Spark es menor o igual a la exactitud de los modelos de SemEval-2017 Task 4: Message Polarity Classificaction (H_0: $\bar{X}_{aot} \leq \mu_{semEval}$), y H1: la exactitud del modelo de análisis de opinión en Twitter por la clasificación al mundial de futbol Rusia 2018 de la selección peruana de futbol con Spark es mayor a la exactitud de los modelos de SemEval-2017 Task 4: Message Polarity Classification (H_1: $\bar{X}_{aot} > \mu_{semEval}$); con un nivel de significancia de 0,05 o 5% de error $\alpha = 0{,}05 = 5\%$, se utilizó la distribución Z: $Z_\alpha = Z_{0,05} = 1{,}64$, donde $n = 813$; n > 30.

Estadística de prueba, Se ha usado los datos proporcionados por la tabla 7 en el remplazo de la ecuación.

$$Z_1 = \frac{679 - 529}{4.83/\sqrt{813}} = 885{,}12$$

Luego de realizada la prueba estadística, rechazamos la hipótesis nula H_0 por lo tanto se acepta la hipótesis alterna H_1, por lo que la exactitud del modelo de análisis de opinión en Twitter por la clasificación al mundial de futbol Rusia 2018 de la selección peruana de futbol con Spark es mayor a la exactitud de los modelos de SemEval-2017 Task 4: Message Polarity Classification, por lo que su eficiencia es adecuado.

Se obtuvo un programa Spark que siguió todas las etapas del método de procesamiento de análisis de opinión (Liu, 2012), se ejecutó con éxito y sin inconvenientes; la exactitud alcanzada por el modelo fue 83,51% lo que significa que el modelo predice correctamente 8 de cada 10 tuits sean estos de clase positivo o negativo, también alcanzó una precisión del 82.30% significando que el modelo predice 8 de cada 10 tuits solo cuando se trata de la clase positiva y finalmente alcanzo una medida del 89.28% en recuperación o sensibilidad, sugiriendo que el modelo predice correctamente casi 9 de cada 10 tuits correctamente de entre todos los casos predichos positivamente sean falsos o verdaderos. Al comparar estos resultados con una marca estándar, se puede observar que son mayores a los obtenidos en la Tarea 1: "Message Polarity Classification" del SemEval-2017 Task 4 (Rosenthal, Farra, & Nakov, 2017), si bien es cierto que estos modelos no han entrenado con el mismo dataset, se puede decir también que las marcas de evaluación alcanzadas por el analizador de opinión en Spark están dentro de las marcas base que cualquier modelo de clasificación binaria debería alcanzar, es decir estar sobre el 70 % en exactitud, indicando que el modelo programado en una arquitectura distribuida se comporta eficientemente como

UNIVERSIDAD NACIONAL DEL ALTIPLANO
REVISTA DE INVESTIGACIONES DE LA ESCUELA DE POSGRADO
Versión electrónica ISSN 2077–8686
Versión Impresa ISSN 1997-4035
ARTÍCULO ORIGINAL

Vol. 9 No.2, pp. 1530-1544, abril/junio 2020
Recibido 05/02/2020
Aceptado 27/04/2020
Publicado 30/04/2020
DOI: http://dx.doi.org/10.26788/riepg.2020.2.169

otros modelos programados en arquitectura centralizada como Sklearn, Numpy o distribuida como TensorFlow (Tabla 7).

Tabla 7. Promedio de tuits correctamente clasificados en función a la exactitud del modelo de análisis de opinión y de SemEval.

Modelo	**Exactitud**	$\bar{X}$ **Tuits correctamente clasificados**
Modelo de Análisis de opinión del microblogging Twitter por la clasificación al mundial de futbol de Rusia-2018.	**83,51%**	**N = 813** tuits del dataset de Prueba; Entonces, $\bar{X}_{aot} = 679$ tuits correctamente clasificados y $\sigma = 4.83$.
Modelos de Regresión Logística para la Tarea 1: Message Polarity classification.	**65,03%**	**N = 813** tuits del dataset de Prueba; Entonces, $\mu_{semEval} = 529$ tuits correctamente clasificados.

Se comprobó la importancia de la etapa 2 del proceso de análisis de opinión, al limpiar el texto de los tuits de cadenas que no contribuyen al discurso de la opinión, sino generan ruido inútil al texto, lo que incluyó la eliminación de signos de puntuación, URLs, hashtags, y nombre de cuentas (Go *et al.*, 2009), aunque en otros modelos no se eliminan estas cadenas propiamente sino se normalizan con palabras remplazantes como URL, REF y TAG dentro del texto (Nodarakis *et al.*, 2016). Además como parte de esta etapa fue necesario estandarizar también todos los caracteres atildados presentes en el texto del tuit ya que son propios del idioma español, que a diferencia del inglés, si pueden generar estimaciones numéricas diferentes para dos palabras que significan los mismo aunque una este escrita con tilde y la otra sin tilde cuando sean vectorizadas numéricamente, lo que conllevo a realizar este paso adicional de remplazo de las vocales atildadas por vocales sin tilde.

El modelo sólo incluyo tokenización en unigramas (una sola palabra) y no en otras combinaciones posibles que otros estudios han probado dentro de sus modelos (Pang & Lee, 2008), (McCreadie *et al.*, 2012) y (McMinn *et al.*, 2013), ya que esta aproximación estuvo fuera del ámbito de la investigación.

La realización de la ingeniería de características fue fácil de realizar en Spark para la eliminación de aquellas palabras sin utilidad compuesta por artículos, preposiciones, y conectores que no agregan valor sino al contrario generan falsos conteos TF-IDF en la vectorización (Liu, 2012) ya que se aplicó la API Stop Word Remover de Spark configurable también para el idioma español (Apache Spark, 2019), la vectorización numérica también fue sencilla de realizar porque se usó la API HashingTF configurada también por defecto para en conteo TF-IDF (Chambers & Zaharia, 2018).

Se eligió Regression Logistic como algoritmo para el modelo de análisis de opinión, este algoritmo de clasificación binaria fue entrenado en 10 repeticiones con un parámetro de regularización de 0,01 de la función objetivo sobre el 70% del dataset consignado como dataset de entrenamiento (Singh, 2018). No se probó otra cantidad de repeticiones y parámetros de regularización en el entrenamiento del modelo porque sólo se pretendió encontrar una marca inicial de comportamiento en la eficiencia del modelo sobre el dataset PeruARusia2018 que carecía de esta medida, por ello se aplicó el algoritmo ideal para análisis de sentimiento recomendado por (Zhai & Massung, 2016).

UNIVERSIDAD NACIONAL DEL ALTIPLANO
REVISTA DE INVESTIGACIONES DE LA ESCUELA DE POSGRADO
Versión electrónica ISSN 2077–8686
Versión Impresa ISSN 1997-4035
ARTÍCULO ORIGINAL

Vol. 9 No.2, pp. 1530-1544, abril/junio 2020
Recibido 05/02/2020
Aceptado 27/04/2020
Publicado 30/04/2020
DOI: http://dx.doi.org/10.26788/riepg.2020.2.169

CONCLUSIONES

El analizador de opinión en Twitter por la clasificación al mundial de fútbol Rusia 2018 de la selección peruana de fútbol con Spark alcanzó una exactitud del 83.51 % usando un modelo de aprendizaje de tipo clasificación binaria basada en Regresión Logística, el cual es significativamente aceptable, además el modelo ha obtenido una precisión de 82.30%, y recuperación o sensibilidad del 89.28%.

El dataset de tuits construido "PeruARusia2018.csv" para realizar análisis de opinión en Twitter por la clasificación al mundial Rusia-2018 de la selección peruana de futbol es adecuado para entrenar el modelo de aprendizaje de tipo clasificación binaria, el cual cumplió con 9 de 12 características aceptables que los datasets estándares que la comunidad científica utiliza en análisis de opinión o sentimiento.

El pre procesamiento del dataset, entrenamiento y evaluación del modelo de análisis de opinión se realizó completamente dentro de Spark, el que incluyó la limpieza total del texto de cada tuit etiquetado antes que sea tokenizado, vectorizado, entrenado y finalmente evaluado.

LITERATURA CITADA

Aguilar, L. J. (2016). Big Data, Análisis de grandes volúmenes de datos en organizaciones: Alfaomega Grupo Editor.

AI Zone, D. (2019). Streaming ML Pipeline for Sentiment Analysis Using Apache APIs: Kafka, Spark, and Drill (Part 1). Recuperado Jun 2, 2019, de https://dzone.com/articles/streaming-machine-learning-pipeline-for-sentiment

Apache Spark, o. (2019). MLlib Main Guide Spark 2.3. Recuperado Ago 15, 2019, de https://spark.apache.org/docs/2.3.0/ml-guide.html

Baltas, A., Kanavos, A., & Tsakalidis, A. K. (2016). *A*n apache spark implementation for sentiment analysis on twitter data. Paper presented at the International Workshop of Algorithmic Aspects of Cloud Computing.

Chambers, B., y Zaharia, M. (2018). Spark: the definitive guide: big data processing made simple: " O'Reilly Media, Inc.".

Fifa.com. (2018). 2018 FIFA world cup RUSSIA all matches in southamerica. Recuperado Oct 08, 2018, de https://www.fifa.com/worldcup/preliminaries/southamerica/

Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford, 1*(12), 2009.

Jefferson, H. (2016). GetOldTweets Programatically. Recuperado Oct 18, 2018, de https://github.com/Jefferson-Henrique/GetOldTweets-python

Leskovec, J., Rajaraman, A., y Ullman, J. D. (2014). Mining of massive datasets: Cambridge university press.

Liu, B. (2011). Web data mining: exploring hyperlinks, contents, and usage data: Springer Science & Business Media.

Liu, B. (2012). Opinion mining and sentiment analysis *Web Data Mining* (pp. 459-526): Springer.

Vol. 9 No.2, pp. 1530-1544, abril/junio 2020
Recibido 05/02/2020
Aceptado 27/04/2020
Publicado 30/04/2020
DOI: http://dx.doi.org/10.26788/riepg.2020.2.169

McCreadie, R., Soboroff, I., Lin, J., Macdonald, C., Ounis, I., y McCullough, D. (2012). *On building a reusable Twitter corpus.* Paper presented at the Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval.

McMinn, A. J., Moshfeghi, Y., y Jose, J. M. (2013). Building a large-scale corpus for evaluating event detection on twitter. Paper presented at the Proceedings of the 22nd ACM international conference on Information & Knowledge Management.

Nodarakis, N., Sioutas, S., Tsakalidis, A. K., & Tzimas, G. (2016). Large Scale Sentiment Analysis on Twitter with Spark. Paper presented at the EDBT/ICDT Workshops.

Pang, B., y Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, *2*(1–2), 1-135.

Rosenthal, S., Farra, N., & Nakov, P. (2017). SemEval-2017 task 4: Sentiment analysis in Twitter. Paper presented at the Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017).

Singh, P. (2018). Machine Learning with PySpark: With Natural Language Processing and Recommender Systems: Apress.

Svyatkovskiy, A., Imai, K., Kroeger, M., y Shiraito, Y. (2016). Large-scale text processing pipeline with Apache Spark. Paper presented at the 2016 IEEE International Conference on Big Data (Big Data).

Tan, P.-N., Steinbach, M., & Kumar, V. (2006). Introduction to data mining: Pearson Education, Inc.

Trendogate.com. (2018). Twitter Trends Archive. Recuperado, de https://trendogate.com/

Twitter.com. (2018). Documentación de la API Rest de Twitter. Recuperado, de https://developer.twitter.com/en/docs

Witten, I. H., Frank, E., Hall, M. A., y Pal, C. J. (2016). Data Mining: Practical machine learning tools and techniques: Morgan Kaufmann.

Zafarani, R., Abbasi, M. A., y Liu, H. (2014). Social media mining: an introduction: Cambridge University Press.

Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., . . . Franklin, M. J. (2016). Apache spark: a unified engine for big data processing. Communications of the *ACM*, *59*(11), 56-65.

Zhai, C., & Massung, S. (2016). Text data management and analysis: a practical introduction to information retrieval and text mining: Morgan & Claypool.