UNIVERSIDAD NACIONAL DEL ALTIPLANO

ESCUELA DE POSGRADO

MAESTRÍA EN INFORMÁTICA

TESIS

ANÁLISIS DE OPINIÓN DEL MICROBLOGGING TWITTER POR LA CLASIFICACIÓN AL MUNDIAL DE FÚTBOL RUSIA - 2018 DE LA SELECCIÓN PERUANA DE FÚTBOL, USANDO EL FRAMEWORK SPARK

PRESENTADA POR:

MAYENKA FERNANDEZ CHAMBI

PARA OPTAR EL GRADO ACADÉMICO DE:

MAGISTER SCIENTIAE EN INFORMÁTICA

MENCIÓN EN INGENIERÍA DE SOFTWARE

PUNO, PERÚ

2019

UNIVERSIDAD NACIONAL DEL ALTIPLANO

ESCUELA DE POSGRADO

MAESTRÍA EN INFORMÁTICA

TESIS

ANÁLISIS DE OPINIÓN DEL MICROBLOGGING TWITTER POR LA CLASIFICACIÓN AL MUNDIAL DE FÚTBOL RUSIA - 2018 DE LA SELECCIÓN PERUANA DE FÚTBOL, USANDO EL FRAMEWORK SPARK

PRESENTADA POR:

MAYENKA FERNÁNDEZ CHAMBI

PARA OPTAR EL GRADO ACADÉMICO DE:

MAGISTER SCIENTIAE EN INFORMÁTICA

MENCIÓN EN INGENIERÍA DE SOFTWARE

APROBADA POR EL SIGUIENTE JURADO:

PRESIDENTE

.....
Mg. MARCO ANTONIO RAMOS GONZALES

PRIMER MIEMBRO

.....
Dr. RUDY ALVARO ARPASI PANCCA

SEGUNDO MIEMBRO

.....
M.Sc. NESTOR TIPULA QUISPE

ASESOR DE TESIS

.....
Dr. JORGE LUIS APAZA CRUZ

Puno, 06 de diciembre de 2019

ÁREA: Ingeniería de Software.
TEMA: Minería de Texto.

DEDICATORIA

En memoria de mi amado padre Oscar Fernández Villa, mi querida mamá Nolberta Chambi Nuñez, y mis hermanos Elmer, Yomira, Nevenka y Giancarlo les dedico mi trabajo de investigación, por protegerme y apoyarme siempre, con ustedes a mí lado soy cada día mejor.

A mi estimada amiga Alodia, mis logros también son de ella.

Y finalmente, a mis niños de mirada leal y aroma a cariño eterno, su afecto me completa.

AGRADECIMIENTOS

- A la Universidad Nacional de Altiplano, a través de la escuela de Pos Grado por los conocimientos vertidos en mí y la oportunidad de permitirme el crecimiento profesional.
- A los miembros del jurado Mg. Marco Antonio Ramos Gonzales, Dr. Rudy Álvaro Arpasi Pancca, por sus correcciones en perfeccionar esta investigación.
- A mi asesor Dr. Jorge Luis Apaza Cruz por las sugerencias y consejos en el proceso de elaboración de esta investigación.
- A Jefferson Henrique, por proporcionar la logística de arrastre de tuits históricos sin el uso directo de la API de Twitter.
- A Databricks, por los tutoriales en Data Science, Big Data y uso masivo de analítica de datos con Spark.
- A Coursera inc, por los cursos abiertos en Data Science, Big Data y Computación Distribuida de las universidades más prestigiosas, gracias.
- A Internet Arxive, por poner a disposición para todos, los textos de vanguardia, gracias.

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE ANEXOS

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 3000 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 3000 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.

Key words: Big Data, MLlib, opinion analysis, social network, Spark, text classification.

INTRODUCCIÓN

El análisis de opinión o análisis sentimental se fundamenta en la era de la Web 2.0 en el que los usuarios de la Web tienen la capacidad de producir contenido en forma de texto como un sensor subjetivo dentro de las aplicaciones Web, al comienzo en los blogs donde el usuario comenta u opina de lo que quiera, luego en las plataformas de comercio electrónico donde el usuario opina o comenta sobre lo que compra o toma un servicio, p.ej. las revisiones de ítems de la tienda en línea amazon.com, y finalmente en las plataformas sociales donde el usuario opina o comenta sobre lo que siente en forma instantánea y abundante, a niveles de Big Data, p.ej. Twitter, Facebook e Instagram; natural y orgánicamente impulsadas por el éxito de los dispositivos móviles.

Es entonces que la minería de datos, y específicamente el análisis de texto o minería de texto toma relevancia y se afinan métodos para tratar de comprender las opiniones escritas y producidas por los usuarios. Ya que el ser humano toma sus decisiones en base o bajo la influencia de las opiniones de sus pares, la minería de opinión o análisis sentimental es importante y relevante, ya que sus modelos permiten entender el subjetivismo presente de los que publican y así pueden ser usados como insumos para sistemas de marketing digital, sistemas de recomendación, e incluso en inteligencia de negocios aprovechando la gran cantidad existente de publicaciones en las plataformas sociales como Twitter.

El objetivo del presente estudio estuvo orientado en clasificar sentimentalmente los tuits publicados en Twitter durante el periodo de clasificación del equipo peruano de futbol al mundial Rusia-2018 bajo una perspectiva distribuida y escalable para demostrar que los modelos de minería de texto utilizando aprendizaje de máquina se pueden desarrollar en sistemas computacionales unificados de Big Data como Spark.

Este trabajo se desarrolló en cuatro capítulos que se detalla a continuación:

En el capítulo I, se expone el marco teórico de la investigación, y los antecedentes del tema; el marco teórico está compuesto por minería de datos, minería web, minería de texto, análisis de opinión, métodos de aprendizaje supervisado para análisis de opinión, evaluación de modelos de aprendizaje supervisado, construcción de corpus o *dataset* de texto, procesamiento de datos, y Spark compuesto por su arquitectura, estructura de datos, pipeline, transformadores, estimadores y la librería MLlib.

El capítulo II, se desarrolla el planteamiento y formulación del problema, además se establece la importancia y relevancia de esta investigación, los objetivos que se han pretendido alcanzar y la hipótesis de estudio.

El capítulo III, muestra el diseño de investigación, la metodología y los materiales utilizados.

El capítulo IV, contiene el flujo de trabajo del análisis de opinión consistente en: recolección de datos y construcción del corpus o *dataset*, preparación de datos, aplicación del modelo de minería de datos de clasificación de texto binario, y evaluación del modelo dentro el *framework* distribuido Spark.

Finalmente se muestra los resultados y conclusiones a los que ha llegado esta investigación, así como las recomendaciones para futuras investigaciones

CAPÍTULO I

REVISIÓN DE LITERATURA

1.1. Marco teórico

1.1.1. Minería de datos

La minería de datos es el proceso de automáticamente descubrir información útil en grandes repositorios de datos. Las técnicas de minería de datos son desplegadas para limpiar grandes bases de datos a fin de hallar patrones novedosos y útiles que podrían de otra manera permanecer desconocidos. Además proporcionan capacidades para predecir la salida de una observación futura, tal como predecir si un nuevo cliente gastará más de 100 dólares en una tienda departamental. Sin embargo no todas las tareas de descubrimiento de información están consideradas como de minería de datos. Por ejemplo, buscar registros individuales usando un sistema de administración de bases de datos o hallar páginas Web vía una consulta en el motor de búsqueda de internet son tareas relacionadas al área de recuperación de la información (Tan *et al.*, 2006).

La minería de datos es el también llamado descubrimiento de conocimiento en bases de datos o Knowledge Discovery in Databases KDD, definido comúnmente como el proceso de descubrimiento de **patrones o conocimiento** útiles a partir de los datos, p.ej. bases de datos, textos, imágenes, la Web y otros. Los patrones deben ser válidos, potencialmente útiles, y entendibles. Así, la minería de datos es un campo multidisciplinario que involucra al aprendizaje de máquina, la estadística, bases de datos, inteligencia artificial, recuperación de información y visualización (Liu, 2011).

La minería de datos en un sentido exclusivo en las redes sociales se define como la *disciplina que provee las herramientas necesarias para descubrir patrones en los*

datos, y así superar los desafíos de analizar grandes cantidades de datos sin procesamiento o *raw* data generados diariamente por los individuos en las redes sociales, alrededor de 6 billones de fotos subidos mensualmente en Facebook, 72 horas de video subidos cada minuto a YouTube, y más de 40 millones de tuits publicados diariamente. Con estos rangos sin precedentes de generación de contenido, los individuos son fácilmente superados por los datos y la dificultad de descubrir contenido relevante de sus intereses son tareas que solamente con herramientas de minería de datos pueden ser superadas (Zafarani *et al.*, 2014).

La minería de datos en un sentido de procesamiento de grandes volúmenes de datos es el *descubrimiento de modelos* para los datos. Los modelos pueden estar basados en modelos estadísticos, modelos de aprendizaje de máquina, modelos basados en soluciones computacionales que modelen los datos, modelos basados en resúmenes, modelos basados en extracción de características (Leskovec *et al.*, 2014).

Y cuando hablamos de Big Data, la minería de datos se refiere al proceso de buscar información valiosa del negocio en una base de datos, *data warehouse* o *data mart*. La minería de datos es un proceso que utiliza técnicas estadísticas, matemáticas, inteligencia artificial y de aprendizaje de máquina para extraer e identificar información útil que convierte en conocimiento a partir de grandes bases de datos, *data warehouse* o *data mart*. Esta información incluye patrones normalmente extraídos de un conjunto grande de datos. Estos patrones pueden ser reglas, afinidades, correlaciones, tendencias o modelos de predicción. Dentro de las categorías de minería de datos, además de la generalista están: la **minería Web**, para la búsqueda y análisis de información en la Web; la **minería de texto** y otros formatos de medio, y permiten descubrir la opinión o el sentimiento incrustado, por ejemplo, en mensajes de texto, en *posts* de Twitter (Aguilar, 2016).

1.1.1.1. El proceso de la minería de datos

La minería de datos es una parte integral de **KDD** que es el proceso general de convertir datos sin procesamiento en información útil (Tan *et al.*, 2006). La Figura 1 esquematiza este proceso.

Los datos de entrada pueden estar almacenados en una variedad de formatos: archivos planos, hojas de cálculo, o tablas relacionales; y pueden residir en un

repositorio de datos centralizado o estar distribuidos entre múltiples sitios. El propósito del **pre procesamiento** es transformar los datos de entrada sin procesar en un formato apropiado para el subsecuente análisis. Los pasos involucrados en el pre procesamiento de datos incluyen fusionar datos a partir de múltiples recursos, limpiar datos para quitar ruido y observaciones duplicadas, y seleccionar los registros y características que son relevantes para la **tarea de minería de datos** en mano. Debido a las varias formas de datos que pueden ser recolectados y almacenados, el pre procesamiento de datos es tal vez el paso más laborioso y consumidor de tiempo en el proceso general de descubrimiento de conocimiento. Medidas estadísticas o métodos de evaluación de hipótesis son aplicados durante el pos procesamiento para eliminar resultados esporádicos de la minería de datos.

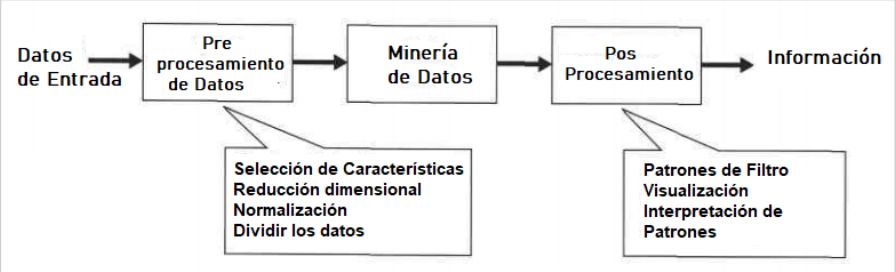


Figura 1. Proceso General de Descubrimiento de Conocimiento en Bases de Datos KDD
Fuente: Adaptado de (Tan *et al.*, 2006)

La Figura 2 muestra el proceso de minería de datos en función a la extracción de patrones de KDD (Zafarani *et al.*, 2014), este proceso describe que se toma los datos sin procesamiento como entradas y se suministra como salida conocimiento patrones con significado estadístico hallados en la entrada. Así, a partir de los datos sin procesamiento, un *subconjunto* es seleccionado para que sea procesado y se le denota como *datos objetivo*. Los datos "objetivo" son *pre procesados* para que estén listos para el análisis usando algoritmos de minería de datos. La minería de datos es luego aplicado en los datos pre procesados y transformados para extraer patrones interesantes. Los patrones son *evaluados* para asegurar su validez y robustez e *interpretados* para solventar valor dentro de los datos.

El proceso completo descrito en la Figura 1 es siempre iterativo. Toma muchas rondas para lograr el resultado satisfactorio para que sea incorporado a tareas del mundo real.

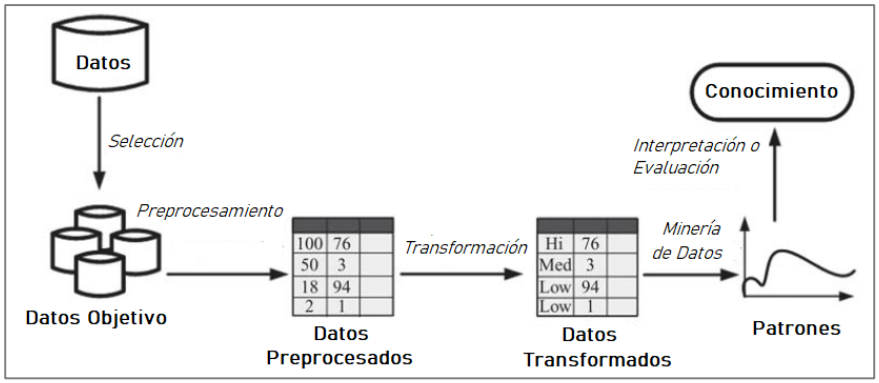


Figura 2. Proceso de Descubrimiento de Conocimiento según KDD

Fuente: Adaptado de (Zafarani *et al.*, 2014)

Finalmente, la minera de datos propiamente en las **redes sociales**, enfrentan la vasta cantidad de datos sin procesamiento que son el contenido generado por los individuos, y el conocimiento que engloba patrones interesantes observados en estos datos. Por ejemplo, para una tienda en línea que vende libros, los datos sin procesar están compuestos por la lista de libros que los individuos compran, y un patrón interesante que hallar podría ser describir los libros que los individuas a menudo compran (Zafarani *et al.*, 2014).

1.1.1.2. Los desafíos que evolucionan la minería de datos

Los actuales desafíos específicos que motivan el desarrollo de la minería de datos se exponen como (Tan *et al.*, 2006):

Escalabilidad, debido a los avances en la generación y recolección de datos, los conjuntos de datos con tamaños de gigabytes, terabytes, o incluso petabytes ahora son muy comunes. Los algoritmos de minería de datos deben manejar los conjuntos de datos masivos, y ser escalables. Por ejemplo, algoritmos fuera del núcleo serán necesarios cuando se procesen conjuntos de datos que no se ajusten dentro de la memoria, así la escalabilidad puede ser mejorada usando muestras o desarrollo paralelo y algoritmos distribuidos.

Alta dimensión, ahora es frecuente encontrar conjuntos de datos con cientos o miles de atributos en vez del puñado normal de hace pocas décadas atrás. Así,

las técnicas de análisis que fueron desarrolladas para datos de pocas dimensiones no pueden funcionar bien en las actuales con datos de alta dimensión (incremento del número de características).

Datos heterogéneos y complejos, se necesitan técnicas que puedan manejar los atributos heterogéneos actuales, como el de las páginas web que contienen texto semi estructurado e hiperenlaces que hallen relaciones entre los elementos en los datos de texto semi-estructurados.

Propiedad y distribución de los datos, a veces los datos necesarios para el análisis no están almacenados en una locación o le pertenecen a la organización. Por ello se necesita de técnicas de "minería de datos distribuida". Entre los desafíos clave que enfrentan los algoritmos de minería de datos distribuidos son: (1) como reducir el número de comunicación necesaria para realizar la computación distribuida, (2) como efectivamente consolidar los resultados obtenidos de la minería de datos a partir de múltiples recursos, y (3) como direccionar las cuestiones relacionadas con la seguridad de los datos.

Análisis no tradicional, las soluciones estadísticas tradicionales están basadas en el paradigma de hipótesis y prueba. Desafortunadamente, estas soluciones son extremadamente laboriosas, y las tareas de análisis de datos recientes requieren de la generación y evaluación de cientos de hipótesis, y consecuentemente, el desarrollo de algunas técnicas están motivadas por la automatización del proceso de la generación y evaluación de la hipótesis.

1.1.1.3.Tareas de la minería de datos

Las tareas de minería de datos se dividen generalmente en dos categorías (Tan *et al.*, 2006):

Tareas Predictivas, el objetivo de estas tareas es predecir el valor de un atributo particular basado en los valores de otros atributos. El atributo a ser pronosticado es conocido como *variable dependiente u objetivo*, mientras que los atributos usados para realizarlo son conocidos como *variables independientes o explicativas.*

Tareas Descriptivas, el objetivo es el patrón derivado que resume la relación subyacente en los datos (correlación, tendencias, agrupación, y anomalías). La Figura 3 muestra cuatro de las tareas principales de la minería de datos.

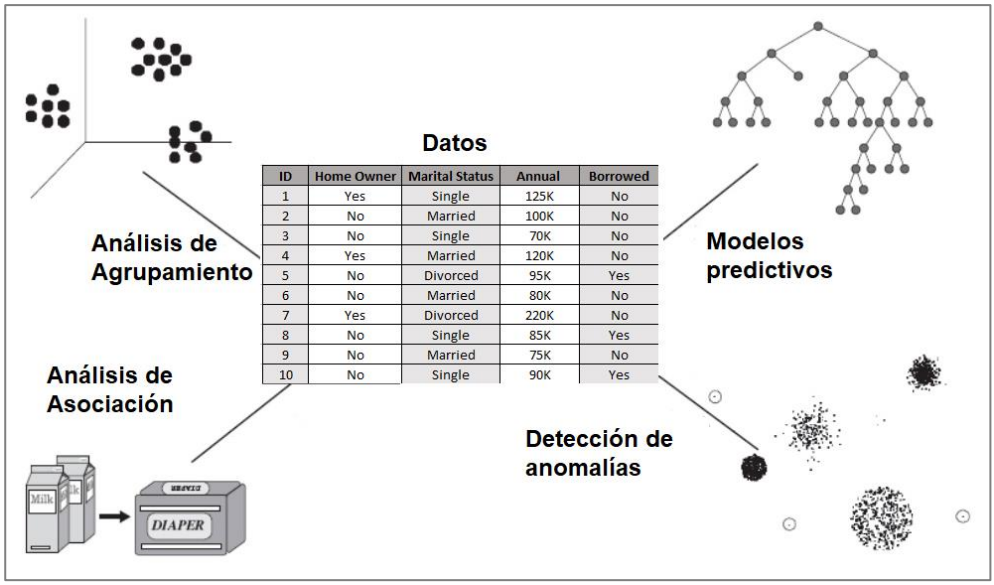


Figura 3. Cuatro de las tareas principales de minería de datos

Fuente: Adaptado de (Tan *et al.*, 2006)

Como lo muestra la Figura 3 las tareas de minería de datos principales son el modelamiento predictivo, análisis de asociación, análisis de agrupamiento y detección de anomalías.

Los modelos predictivos se refieren a las tareas de construir un modelo para la variable objetivo como una función de las variables explicativas; hay dos tipos de tareas de modelamiento predictivo que son la **clasificación y la regresión**. La clasificación se usa en variables objetivo discretas, mientras que la regresión en variables objetivo continuos. El objetivo de ambas tareas es aprender un modelo que minimice el error entre los valores verdaderos y pronosticados de la variable objetiva. Por ejemplo, la predicción si un usuario de la Web realizará una compra en una tienda de libros en línea es una tarea de clasificación ya que la variable objetivo es de valor binario.

El análisis de asociación se usa para descubrir patrones que describan fuertes asociaciones en los atributos de los datos. El descubrimiento de patrones está típicamente representados en la forma de reglas de implicación o subconjuntos de atributos. Debido al tamaño exponencial en el espacio de búsqueda, el

objetivo del análisis de asociación es extraer los patrones más interesantes en una forma eficiente. Por ejemplo, identificar páginas Web que son accedidas en conjunto.

El análisis de agrupamiento busca hallar grupos de observaciones cercanamente relacionados de tal forma que esas observaciones que recaen en el mismo grupo son más similares a cada una de ellas que aquellas que pertenecen a otros grupos. Por ejemplo, hallar los grupos de clientes relacionados.

La detección de anomalía es la tarea de identificar observaciones cuyas características son significativamente diferentes del resto de los datos. Tales observaciones son conocidas como anomalías o valores atípicos. El objetivo del algoritmo de detección de anomalías es descubrir las anomalías reales y evitar falsamente etiquetar objetos normales como anómalos. En otras palabras, un buen detector de anomalía tiene que tener una alta tasa de detección y baja tase de falsas alarmas. Por ejemplo, la detección de fraude.

1.1.2. Minería web

La minería web pretende descubrir la información útil o conocimiento desde las estructuras de los hiperenlaces Web, el **contenido de las páginas**, y los datos de uso común. Aunque la minería Web usa varias técnicas de la minería de datos tradicional no es una aplicación pura de la minería de datos tradicional debido a la naturaleza heterogénea, casi estructurada o no estructurada de los datos en la web. Así la minería de datos tradicional usa siempre estructuras de datos almacenados en tablas relacionales, hojas de cálculo, o archivos sueltos en forma tabular. Con el crecimiento de la web y los documentos de texto, **la minería web** está llegando a ser cada vez más importante y popular (Liu, 2011).

La minería web es como la minería de texto pero que toma las ventajas de la información extra y a menudo mejora los resultados al capitalizar los directorios de tópicos y otra información de la web, que es un repositorio masivo de texto. Que a diferencia de un texto normal, contiene marcados estructurales explícitos, internos que indica la estructura de la página y externos que define enlaces explícitos de hipertexto entre documentos. Ambos apalancan la minera web (Witten *et al.*, 2016).

La minería web es actualmente un área de la minería de datos relacionado a la **información disponible en internet**. Es un concepto de extraer datos informativos disponibles en las páginas web de internet; donde los usuarios usan diferentes motores de búsqueda para obtener los datos requeridos de internet, y esos datos y necesidades informativas se descubren a través de las técnicas de la minería web. Se utilizan diferentes herramientas y algoritmos para la extracción de datos de páginas web que incluyen documentos web, imágenes, etc. La minería web se está volviendo muy importante debido al aumento del tamaño de los documentos de texto en internet y la búsqueda de patrones relevantes, conocimiento y datos informativos que es muy difícil de obtener manualmente. A través de la minería Web se recopila información referente a la estructura o enlaces, uso de páginas visitadas, uso de datos, y contenido como documentos de texto, y páginas (Mughal, 2018).

El término World Wide Web está relacionado a la combinación de documentos web, videos, audios; algunos procesos incluidos en la minería Web son:

Recuperación de la Información, es el proceso de recuperar información relevante y útil sobre la web. La recuperación de la información se ha enfocado más en la selección de datos relevantes a partir de grandes colecciones de datos de bases de datos y descubrir nuevo conocimiento de la gran cantidad de datos para responder a las consultas de los usuarios. Los pasos de la RI incluyen búsqueda, filtros y coincidencias (Svyatkovskiy *et al.*, 2016).

Extracción de información (EI), es un proceso automático de extraer datos analizados (estructurado). EI es una tarea que trabaja como la recuperación de información pero se enfoca más en extraer hechos relevantes.

Aprendizaje de máquina, es un proceso de soporte que ayuda a minar la web. El aprendizaje de máquina puede mejorar la búsqueda en la web conociendo el comportamiento del usuario (interés). Diferentes métodos de aprendizaje de máquina son usados en motores de búsqueda para suministrar servicio web inteligente, p.ej. recuperación de la información. Este es el proceso que tiene la habilidad de aprender el comportamiento del usuario y enriquecer el funcionamiento de una tarea específica.

La minería web puede ser categorizada en tres tipos: minería de la estructura de la web, **minería de contenido** de la web, y minería del uso de la web (Liu, 2011) y (Mughal, 2018). La Figura 4 muestra la taxonomía de minería web.

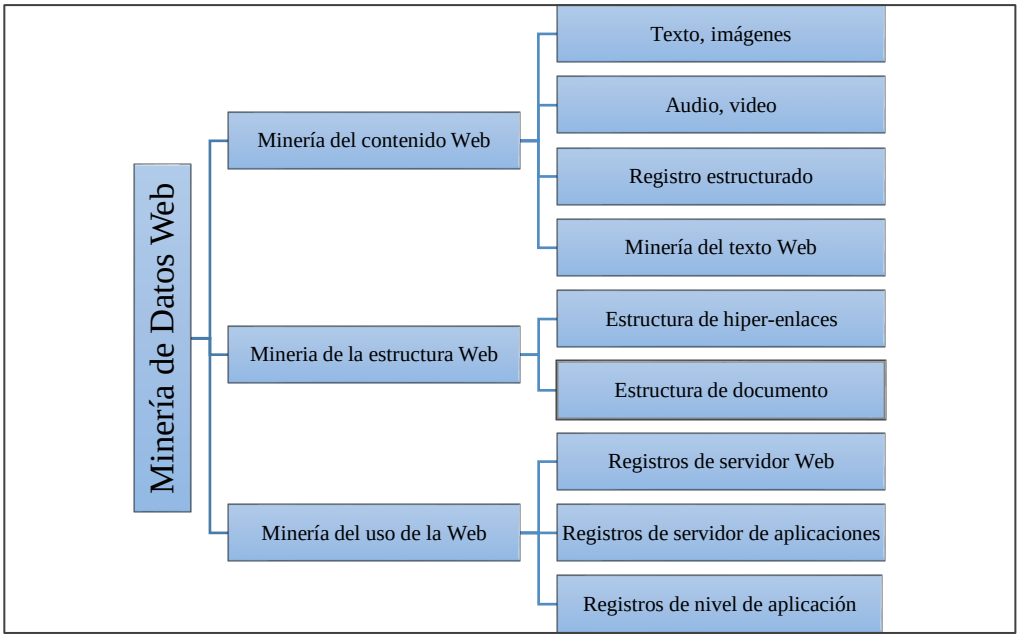


Figura 4. Taxonomía de la minería Web

Fuente: Adaptado de (Mughal, 2018)

A continuación se detalla cada una de las áreas de la minería web.

1.1.2.1.Minería de contenido web

Es un proceso de la minería web en que se extrae información útil de los contenidos dentro de la página web o sitios web (www). El contenido incluye audio, video, documentos de texto, hiperenlaces, y registros estructurados. El contenido web está diseñado para llevar datos a los usuarios en la forma de texto, listas, imágenes, videos y tablas. En la pasada década el número de páginas web (HTML) se incrementó a billones y aún continúa creciendo. La consulta de búsqueda en billones de documentos web es una tarea muy difícil y consumidora de tiempo.

Técnicas de minería del contenido web

La minería de contenido extrae datos consultados al realizar diferentes técnicas de minería de datos. La Figura 5 muestra estás técnicas.

Las cuatro técnicas de minería de contenido descritas en la Figura 5 son usadas para la minería del contenido web, donde la mayoría de los datos del contenido web se encuentra en forma de texto sin estructura. Para la extracción de datos sin estructura, **la minería de contenido web requiere de la minería de texto** y de las soluciones de la minería de datos. Los documentos en texto están relacionados a la minería de texto, aprendizaje de máquina y lenguaje natural. El propósito principal de la minería de texto es extraer la información previa del contenido de los recursos, así la minería de texto es una parte de la minería del contenido web y por lo tanto diferentes técnicas que se usan en la minería de texto de los contenidos de la web sobre los sitios web/internet para suministrar datos desconocidos como la extracción de información, resúmenes, visualización de la información, rastreo de temas, categorización y agrupamiento.

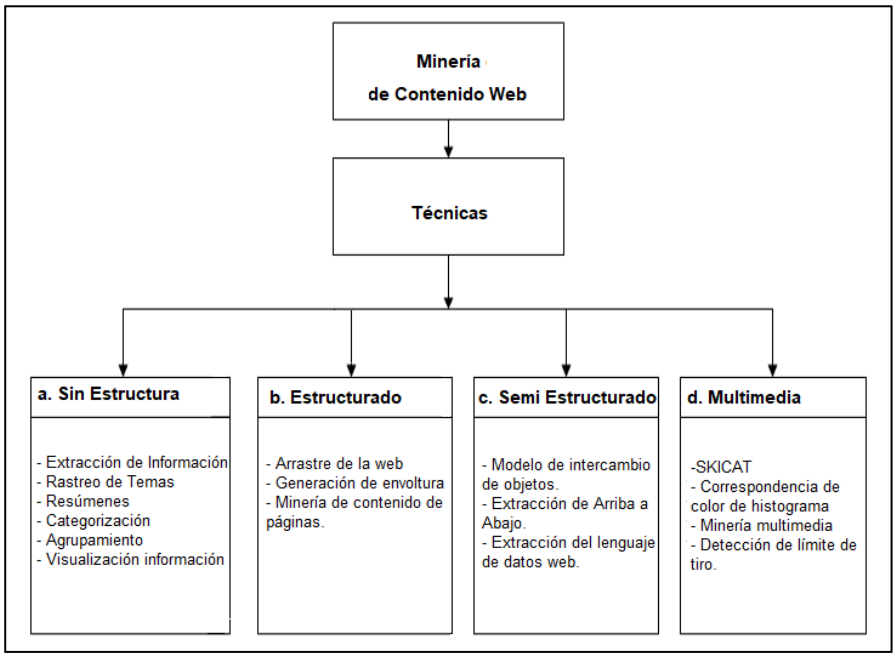


Figura 5. Técnicas de minería del contenido web

Fuente: Adaptado de (Mughal, 2018)

Algoritmos de minería de contenido web

La minería de contenido web usa múltiples técnicas para extraer información de grandes cantidades de datos, hay diferentes tipos de algoritmos que son usados para extraer información conocida, como:

a. Arboles de decisión, es una solución de clasificación basada en estructura que consiste de un nodo raíz, ramas y nodos hojas. Es un proceso jerárquico en el que el nodo raíz es dividido en sub ramas y los nodos hojas contienen las etiquetas de clase. Los arboles de decisión es un técnica muy poderosa.
b. Las redes bayesianas, es un algoritmo simple y poderoso de clasificación basado en el teorema de Bayes. A partir de valores de *datasets* predefinidos, se calculan probabilidades para cada clase al contar las combinaciones de los valores. La clase más probable es la que tiene mayor probabilidad.
c. Máquina de soporte vectorial, es un algoritmo de clasificación de aprendizaje de máquina simple, este método puede ser usado en conjunto de datos lineales y no lineales. El hiperplano de separación óptima es solo una línea que es usada para trazar la separación de dos clases dependiendo de las características diferentes de clasificación.

Redes neuronales, es otra aproximación de minería de contenido web que usa un algoritmo de propagación hacia atrás. El algoritmo consiste de múltiples capas, capa de entrada, algunas capas ocultas y una capa de salida, cada una alimenta la siguiente capa hasta la última. La neurona es la unidad básica de la red neuronal.

La aplicación de la minería web permite que automáticamente se clasifiquen y agrupen páginas web de acuerdo a sus temas. Estas tareas son similares a los de la minería de datos tradicional, sin embargo, se puede descubrir patrones en las páginas web para extraer datos útiles como descripciones de productos, foros de publicación para diversos propósitos. Por lo tanto se puede efectuar tareas de minería en las **revisiones, publicaciones que realizan los usuarios** de la página web y realizar **análisis de opinión** para descubrir las tendencias subjetivas sobre algo o alguien (Liu, 2011).

1.1.2.2.Minería de la estructura web

Descubre conocimiento útil desde los hiperenlaces, que representan la estructura de la web. La minería de la estructura básicamente muestra el resumen estructurado de los sitios web. Identifica la relación entre páginas web enlazadas de los sitios web.

Técnicas de minería de la estructura web

Se usan diferentes técnicas algorítmicas para descubrir datos de la web en el que se analizan los hiperenlaces de los sitios web para recolectar datos informativos y clasificarlos en categorías como similitudes y relaciones. La Figura 6 muestra estas técnicas.

La inter-página es un tipo de minería que se realiza en el nivel de documento y la minería de nivel de hiperenlace. El análisis de enlaces es antigua pero un método muy útil que es aumenta su valor en el área de la investigación de minería web. Por ejemplo, desde los enlaces se puede descubrir páginas web importantes, usado por los motores de búsqueda. También se puede descubrir comunidades de usuarios que comparten intereses comunes.

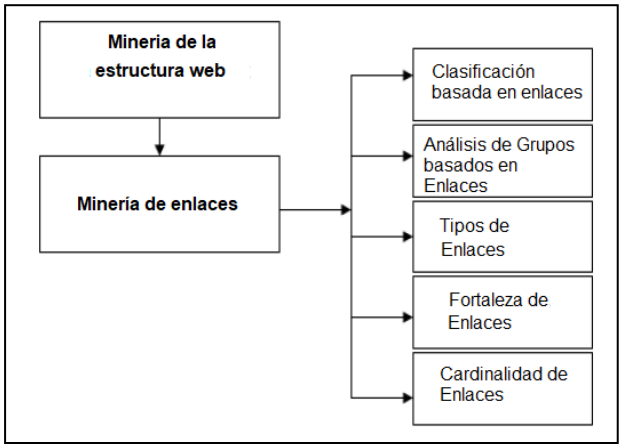


Figura 6. Minería de la estructura de la Web

Fuente: Adaptado de (Mughal, 2018)

Algoritmos de la minería de la estructura web

Hay varios algoritmos de minería de la estructura de la web, como Page Rank, e HITS ambos se enfocan en la estructura de los enlaces de la Web y como dar importancia a las páginas web.

a. Algoritmo Page Rank, fue desarrollado en 1998, por los autores del motor de búsqueda Google. Se examinan los hiperenlaces en las páginas web para sugerir una medida de "prestigio" de cada una de las páginas web y los sitios web. Donde el "prestigio" se define como "alta posición alcanzada a través del éxito o influencia". Page Rank, intentan medir la posición de una página web. Las páginas web que más se enlazan a los sitios web, tienen más prestigio,

especialmente si las páginas están enlazadas a sitios con alto prestigio. Los motores de búsqueda usan PageRank para ordenar las páginas web en orden antes de mostrar los resultados de una búsqueda

b. HITS (Búsqueda de Temas Inducido por Hiperenlaces), es un algoritmo que clasifica páginas en base a dos medidas: autoridades y concentradores. Una página web autoritaria es una página puntuada por altos pesos centrados y los buenos concentradores son las páginas puntuadas por la mayoría de las páginas autoritarias con altos pesos. No es fácil diferenciar estas características en algunos sitios web ya que pueden ser autoritarias o concentradoras al mismo tiempo. El algoritmo incluye dos pasos, primero se saca una muestra en que las páginas relacionadas son recolectadas para ciertas consultas. En pasos iterativos se hallan las páginas autoritativas y concentradoras con la ayuda de las salidas de la muestra. HITS no puede encontrar las páginas relevantes solicitadas por las consultas de los usuarios.

1.1.2.3. Minería del uso de la web

Se refiere al descubrimiento de patrones en el acceso de los usuarios a partir de los registros del uso de la web hallados en servidores web, como por ejemplo los patrones de clic realizados por el usuario en un determinado momento y circunstancias.

La minería del uso de la web es una técnica que automáticamente archiva patrones de acceso de los usuarios y esta información es mayormente suministrada por los servidores web que después son recolectados en registros de acceso. Estos registros son direcciones URL, tiempo de visita, direcciones IP, así se puede observar el comportamiento del usuario en el momento que está interactuando con la web. Hay dos tipos de recolección de patrones general recoge información del historial de la página web y la personalizada de un usuario en específico.

Técnicas de minería de uso de la web

a. Pre procesamiento de datos, los datos del mundo real están incompletos, inconsistentes e ilegible. El pre procesamiento de datos una técnica de minería que integra bases de datos y hace que datos sin procesamiento sea

entendible y consistente. La tarea de pre procesamiento es limpiar, corregir los datos y alistar los datos de entrada para la minería, por ello incluye métodos como limpieza de datos e identificación de usuario y sesión.

La limpieza de datos tiene como propósito remover información irrelevante e innecesaria de los registros, y la técnica de identificación de usuario y sesión es usada para hallas sesiones de usuario de los archivos de acceso de los registros, como la información de inicio de sesión, cookies para identificar ID únicas de visitantes en páginas web específicas. La identificación de sesión es conocer el número de páginas visitadas por un solo usuario en una fila sobre una visita a un sitio web.

b. Descubrimiento de patrones, se usan diferentes técnicas para lograrlo, como por ejemplo: El análisis estadístico que extrae conocimiento sobre las visitas a las páginas web. Se analizan diferentes variables en función a la frecuencia, media, y moda en las sesiones para mostrar el tamaño de la página, accesos de páginas recientes y tiempo de visita.

   Las reglas de asociación ayudan a encontrar correlaciones entre las páginas web que aparecen en la sesión de usuario repetidamente, donde la regla describe la relación entre las páginas visitadas una detrás de otra por los usuarios en el tiempo de su sesión de visita.

   Agrupamiento es un método de agrupamiento de ítems (usuarios y páginas) con similares características en conjunto. La minería de uso consiste en dos tipos de grupos, los grupos de usuarios suministran información sobre el conjunto de usuarios con actividades similares o patrones de búsqueda; los grupos de páginas suministran información sobre páginas web con contenido similares.

   Clasificación es una técnica que clasifica ítems y los mapea en diferentes clases predefinidas, se puede establecer perfiles de usuario.

c. Análisis de patrones, es el último paso de la minería de uso de la web. Ayuda a mejorar el funcionamiento de los sistemas como el uso de agentes inteligentes que detecta elementos recibidos, los reconoce y determina que tarea se debe realizar.

Algoritmos de minería de uso de la web

Los algoritmos más importantes son A priori, Crecimiento FP, y Fuzzy c-means.

a. El algoritmo A priori es un algoritmo supervisado mayormente usado por reglas de asociación para hallar conjuntos frecuentes de ítems durante una transacción, al principio el algoritmo observa bases de datos iniciales y captura aquellos datos que son más grandes, luego usa los resultados para hallar otros conjuntos de datos. El algoritmo predefine un nivel de apoyo mínimo para hallar los conjuntos de ítems que son pequeños y grandes.
b. Crecimiento FP es otro algoritmo eficiente usado en la asociación de reglas, así se descubre conjuntos frecuentes de datos desde árboles FP sin generación de candidato y usa una aproximación de abajo hacia arriba. El árbol FP es una estructura de datos completo, contiene un nodo raíz y sub árboles nodos (prefijo) como hijos. El algoritmo FP busca en el árbol FP y extrae los conjuntos frecuentes de datos.

   Fuzzy c-mean es un algoritmo de agrupamiento no supervisado que aplica un amplio rango de datos conectados. La tarea FCM se encarga de agrupar n objetos en n grupos. Cada grupo tiene un punto central que describe las características e importancia del grupo. Los objetos cercanos al centro del grupo llegan a ser miembros del grupo.

1.1.3. Minería de texto

La minería de texto también llamada minería de datos de texto, que desde un punto de vista práctico es el proceso de deducir información de alta calidad a partir de un texto determinado. Así el análisis de texto trata de encontrar patrones dentro de un conjunto de textos que facilite una mejor toma de decisiones. Además señala que el texto es una de las fuentes de datos más comunes y más grandes de los Big Data, ya que los datos de texto se encuentran en los correos electrónicos, mensajes de texto, tuits, entradas en medios sociales tales como blogs, wikis (posting), mensajes instantáneos (SMS, WhatsApp, GroupMe, Joyn, Viber, Line), chat en tiempo real (Gmail, WhatsApp, Facebook Messenger, Live Messenger), conversión a texto de mensajes de voz, audios, (podcast) faxes y burofaxes, y naturalmente el resto de fuentes de datos como libros, informes, estudios, artículos de prensa, contenidos de sitios web (Aguilar, 2016). La analítica de textos o análisis de textos se enmarca dentro de disciplinas ya muy acreditadas durante años tales como el procesamiento

de lenguaje natural, y la numeración de textos dentro de otras disciplinas como la inteligencia artificial y lingüística computacional como lo muestra la Figura 7.

La minería de texto es hallar patrones dentro del texto. Es el proceso de analizar texto para extraer información que sea útil para propósitos particulares, donde el texto se caracteriza por no tener estructura, es amorfa, y es difícil de tratar con ella, a pesar de ello, en la actualidad, y más en la cultura occidental, el texto es el vehículo más común de intercambio de información formal por lo que la motivación para tratar de extraer información es irresistible incluso si el éxito es sólo parcial. La similitud superficial entre la minería de datos y texto oculta diferencias reales, mientras la minería de datos se caracteriza por la extracción de información implícita, previamente desconocida, y potencialmente útil a partir de los datos, en la minería de texto la información extraída está claramente y explícitamente declarado en el texto. El problema que presente la minería de texto es que la información no está expresado en una manera que sea manejable para el procesamiento automático (Witten *et al.*, 2016).

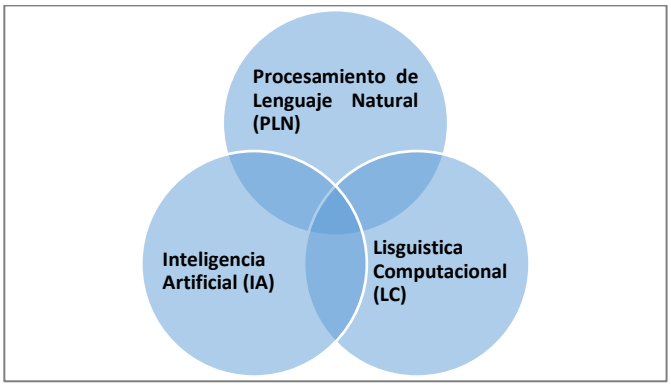


Figura 7. Disciplinas involucradas en la minería de texto

La minería de texto es el análisis de texto que ofrece la oportunidad de descubrir varios tipos de conocimiento útiles para muchas aplicaciones, especialmente conocimiento sobre las opiniones y preferencias de los humanos, que es a menudo directamente expresado en datos de texto. Por ejemplo, ahora es la norma aprovechar dentro de los datos de texto de **opiniones** tales como revisiones de productos, discusión de foros, y texto de las redes sociales para obtener opiniones sobre temas que les interesa y así optimizar varias de las tareas de toma de decisiones tales como comprar un producto o elegir un servicio. Debido a la abrumadora cantidad de información, las personas necesitan herramientas de software inteligente que les ayude a descubrir conocimiento relevante para

optimizar decisiones o para ayudarlos a completar sus tareas más eficientemente (Zhai y Massung, 2016).

Aunque la tecnología que apoya la minería de texto no es aún lo suficiente madura como los motores de búsqueda que soportan el acceso de texto, ha surgido un progreso significativo en los recientes años, y ya se usa herramientas especializadas de minería de texto en varios dominios aplicativos. A diferencia de los datos estructurados que conforman esquemas muy bien definidos y son relativamente más fáciles de computar, el **texto** tiene una estructura menos explícita, por lo que se requiere procesamiento computacional para comprender el contenido codificado en texto. La Figura 8 muestra los dos pasos naturales en el proceso de análisis de cualquier "Big data en Texto", recuperación de la información y minería de texto. El primer paso aplica las técnicas de la recuperación de la información para convertir los datos de texto sin procesamiento del *Big Data* en uno más pequeño pero muy relevante, y luego el paso de la minería de texto que permitirá descubrir conocimiento y patrones.

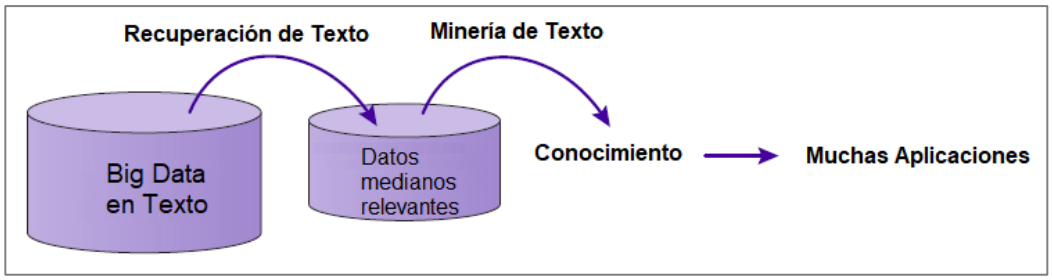


Figura 8. Dos técnicas para analizar Big Data de texto, recuperación y minería

Fuente: Adaptado de (Zhai y Massung, 2016)

Relación entre el texto y los humanos como sensores subjetivos

En el contexto de Big Data, según (Zhai y Massung, 2016) los datos de texto son muy diferentes de otros tipos de datos ya que es generalmente producido por los humanos y a menudo también consumido por los humanos a diferencia de otros datos que tienden a ser generados por las máquinas. Ya que los humanos pueden comprender los datos de texto mejor que las computadoras, el involucramiento del humano en el proceso de minería de texto es absolutamente crucial (más que en otras aplicaciones de Big Data). Se puede comparar a los humanos como sensores subjetivos como lo son los sensores físicos (sensor de red, termómetro). La Figura 9 ilustra esta idea.

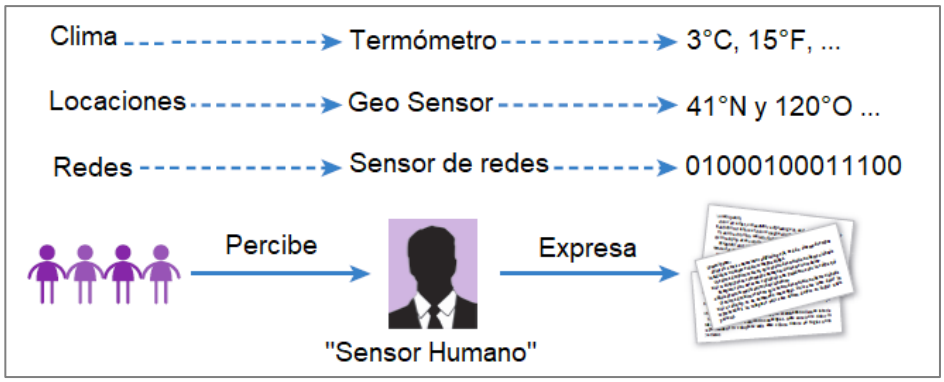


Figura 9. Los humanos como sensores subjetivos

Fuente: Adaptado de (Zhai y Massung, 2016)

Cualquier sensor monitorea el mundo real de alguna forma; entonces el **humano como sensor subjetivo** observa el mundo real desde su propia perspectiva, y expresa lo que ha observado de lo que está aconteciendo en el mundo en forma de texto. Entonces los datos de texto son muy importantes porque contienen conocimiento sobre los usuarios, especialmente preferencias y opiniones.

Al tratar al texto como datos observados a partir de sensores humanos, se puede integrar al *framework* de la minería de datos. La Figura 10 lo expresa.

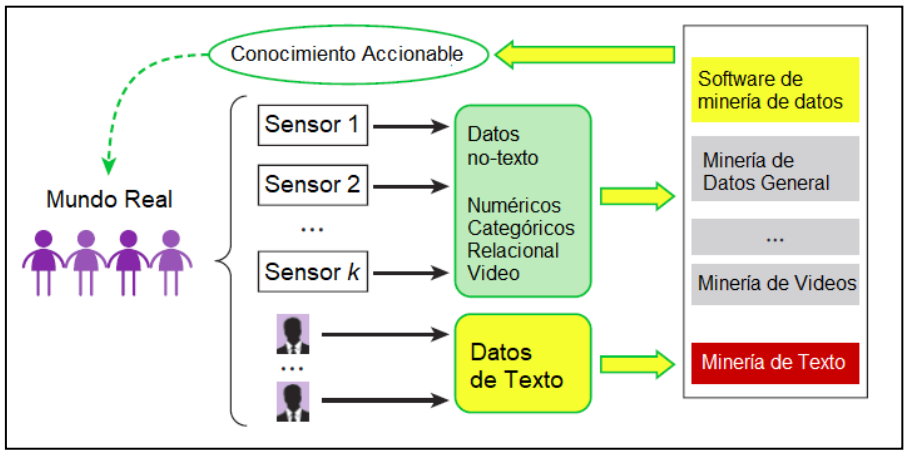


Figura 10. El problema general de minería de datos y de texto

Fuente: Adaptado de (Zhai y Massung, 2016)

Dentro del módulo de la minería de datos se tiene diferentes tipos de algoritmos de minería que se corresponde con los tipos particulares de datos. Por ejemplo los datos en video necesitaran de visión computacional para comprender el contenido del video, que facilitará la efectividad de minería general. Así mismo se necesita de algoritmos que ayuden a convertir datos de texto en conocimiento accesible que se

pueda usar en el mundo real, especialmente en la toma de decisiones. La Figura 11 lo describe.

Panorama de las tareas de la minería de texto

Una descripción de alto nivel del panorama general de varias de las tareas de minería de texto se muestra en la Figura 12; se muestra el proceso de generación de datos de texto en más detalle. Específicamente un humano como sensor o un observador humano examinaría el mundo desde alguna perspectiva. Diferentes personas estarían mirando el mundo desde diferentes ángulos y prestarían atención a diferentes cosas. Los humanos expresan lo que están observando usando un lenguaje natural tal como el español: el resultado es datos de texto. El principal objetivo de la minería de texto es revertir este proceso de generar datos de texto y descubrir varios conocimientos sobre el mundo real como fue observado por el sensor humano (Zhai y Massung, 2016).

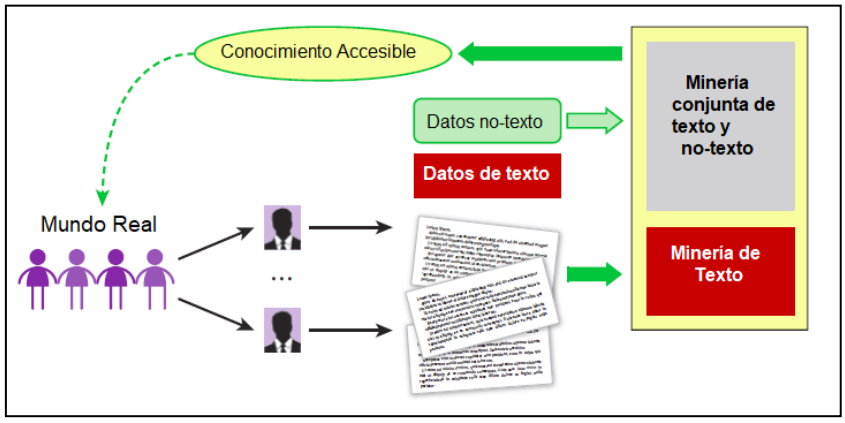


Figura 11. La minería de texto como caso especial de la minería de datos

Fuente: Adaptado de (Zhai y Massung, 2016)

Los cuatro tipos de tareas de minería de texto son:

a. **Minería de conocimiento sobre el lenguaje natural**

Debido a que el texto observado está escrito en un lenguaje en particular, por la minería de texto, se puede minar conocimiento potencial sobre el uso mismo del lenguaje natural. Por ejemplo, si el texto está escrito en inglés, podríamos ser capaces de descubrir conocimiento sobre el inglés, tal como sus usos, direcciones, sinónimos, y coloquialismos.

b. **Minería del contenido de texto**

Se tiene mucho más que hacer con la minería del contenido de los datos de texto, enfocado en extraer las declaraciones más importantes en los datos de texto y convertirlos en información de mayor calidad de un aspecto del mundo del que estemos interesados. Por ejemplo, se puede descubrir todo lo que se ha dicho sobre una persona o entidad en particular. Se puede considerar a la minería de contenido como la descripción del mundo observado en la mente del autor.

c. **Minería de conocimiento sobre el observador**

Debido a que los humanos son como sensores subjetivos, los datos de texto expresados por los humanos a menudo contienen declaraciones subjetivas y opiniones que podrían ser únicos al observador particular humano (productores de texto). Entonces, se puede potencialmente minar los datos de texto para inferir algunas propiedades de los autores que produjeron los datos de texto, tal como el humor o sentimiento de las personas hacia un problema. Se debe distinguir entre la minería de conocimiento sobre el mundo observado de la minería de conocimiento sobre el texto producido ya que los datos de texto generalmente es la mezcla de sentencias objetivas sobre el mundo observado y sentencias subjetivas o comentarios que reflejan las opiniones y creencias de los productores de texto, y es posible y útil extraer cada uno de ellos separadamente.

d. **Inferir conocimiento sobre las propiedades del mundo real**

En la Figura 12, en el lado izquierdo de la figura, se ilustra que la minería de texto puede permitir además inferir valores de variables interesantes del mundo real al influenciar la correlación de los valores de tales variables y el contenido de los datos de texto. Por ejemplo, podría haber alguna correlación entre los cambios del precio de stock del mercado y los eventos reportados en las noticias (p.ej. el reporte de ganancias positivas de una compañía estaría correlacionado con el incremento de los precios de stock de la compañía). Esas correlaciones pueden estar influenciadas para realizar estimaciones basadas en el texto, donde se usa datos de texto como una base de predicción de otras variables que únicamente estarían relacionados remotamente a los datos de texto (p.ej. predicción del precio de stock). La inferencia sobre los factores desconocidos que afectan la toma de decisiones puede tener muchas aplicaciones, especialmente si se puede realizar predicciones sobre los eventos futuros (p.ej. analítica predictiva basada en texto).

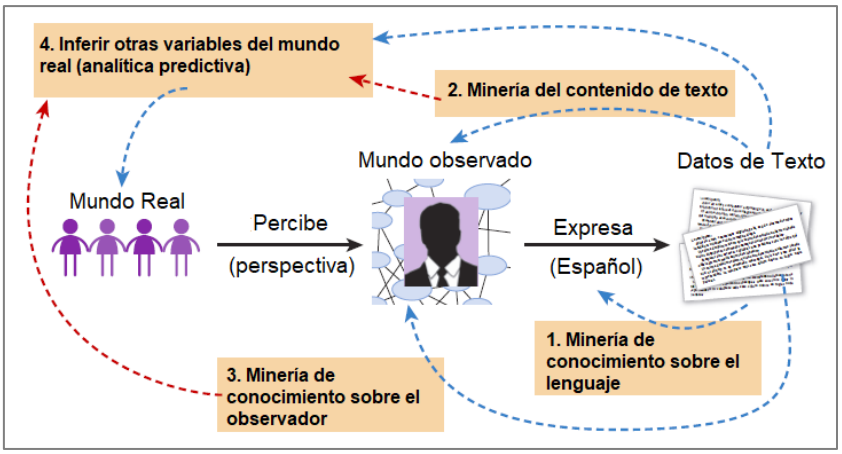


Figura 12. Minería de diferentes tipos de conocimiento de datos de texto

Fuente: Adaptado de (Zhai y Massung, 2016)

1.1.4. Minería de opinión o análisis de sentimiento

El análisis de opinión comienza a marcarse como un área importante de investigación a partir del 2001, (Pang y Lee, 2008) realizan un estudio y señalan que los términos de minería de opinión, análisis de sentimiento y análisis subjetivo están relacionados. Tanto la minería de opinión y análisis de sentimiento son sub áreas del análisis subjetivo, todas tienen como objetivo habilitar a las computadoras el reconocer y expresar opiniones, se pretende diferenciar el lenguaje orientado a la subjetividad a la de la objetiva o factual. La minería de opinión extrae y analiza juicios en varios aspectos de un ítem dado, y el análisis sentimental se enfoca en la aplicación específica de la clasificación de las revisiones, como su polaridad (positivo y negativo). Estas tendencias surgieron por el aumento de métodos de procesamiento del lenguaje natural y recuperación de la información, la disponibilidad de los *datasets* para algoritmos de aprendizaje de máquina, debido a la explosión de la Web.

El análisis de opinión o análisis de sentimiento se define como el campo de estudio que analiza las opiniones, sentimientos, evaluaciones, apreciaciones, actitudes y emociones de las personas hacia las entidades tales como los productos, servicios, organizaciones, individuos, temas y atributos. Las opiniones son importantes para todas las actividades humanas, ya que toda toma de decisión 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 sus productos y/o servicios (Liu, 2012).

En el mundo del Big Data, la minería de opinión o análisis de sentimiento se refiere a la aplicación del procesamiento del lenguaje natural, lingüística computacional y analítica de texto para identificar y extraer información subjetiva de fuentes materiales. El análisis de sentimientos clásico ha sufrido un cambio espectacular desde la implantación de la Web 2.0 y el creciente uso de los blogs y redes sociales. Y en la actualidad el análisis de opinión es de uso popular en el análisis de texto para examinar y obtener la dirección general de la opinión a través de un número grande de personas que proporcionan información sobre lo que el mercado está diciendo, pensando y sintiendo acerca de una organización o persona (Aguilar, 2016).

La minería de opinión y análisis de sentimiento hace uso de los datos en texto que son generados por los humanos como sensores subjetivos, a diferencia de otros tipos de datos como los videos, los datos en texto están enriquecidos con opiniones, y el contenido tiende a ser subjetivo como se muestra en la Figura 13. Este tipo de datos es actualmente una ventaja única de los datos de texto si se les compara con otros datos porque ofrece la gran oportunidad de comprender a los observadores, así como realizar minería de texto para comprender sus opiniones (Zhai y Massung, 2016).

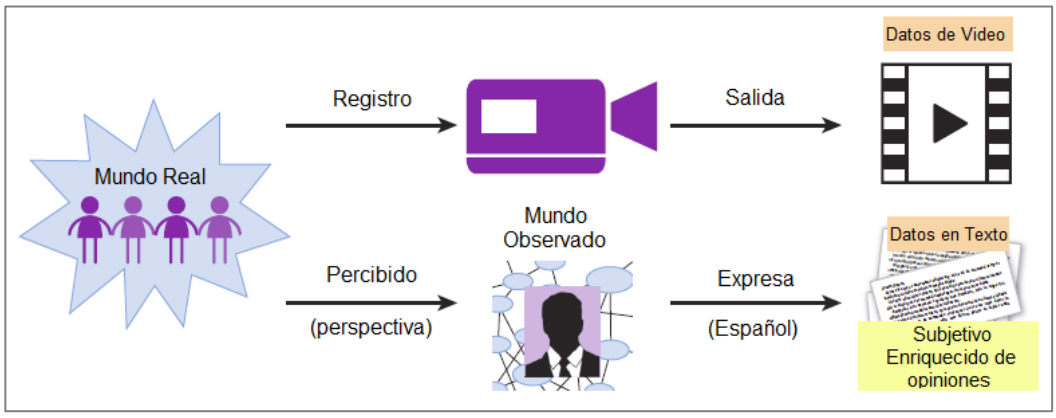


Figura 13. Diferencia entre Sensor objetivo o factual y Sensor subjetivo

Fuente: Adaptado de (Zhai y Massung, 2016)

Opinión

Una opinión es una declaración subjetiva que describe lo que una persona cree o piensa sobre algo, como lo muestra la Figura 14. La palabra **subjetiva** de la figura es un factor diferenciador importante de aquellas declaraciones objetivas o factuales ya que tienden a ser difíciles de probar si son correctas o incorrectas debido a que reflejan lo que la persona piensa sobre algo. En cambio la característica objetiva o factual se prueba en correcta e incorrecta. Por ejemplo, la declaración "la computadora tiene una pantalla y batería" puede ser revisada y verificada si en efecto tiene una pantalla y batería, sin embargo si la declaración fuera "esta computadora tiene la mejor batería" o "la computadora tiene una pantalla bonita" al ser subjetivas es mucha más difícil de probar si son correctas o incorrectas.

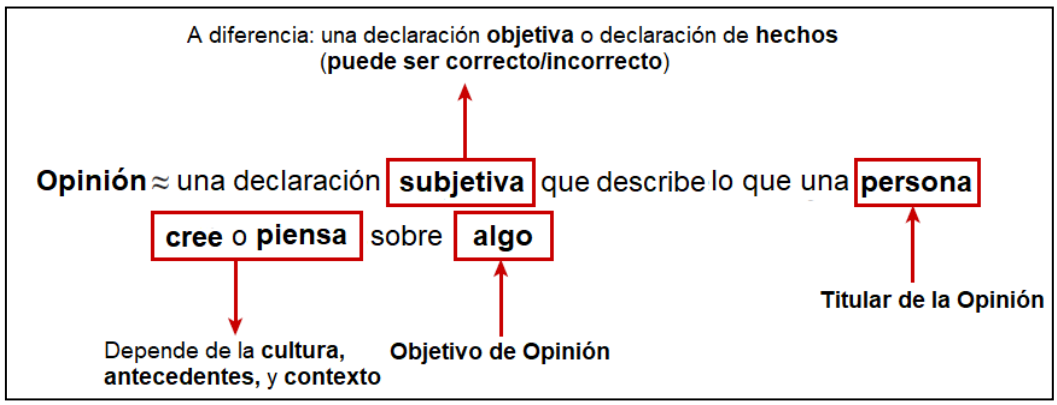


Figura 14. Definición de opinión

Fuente: Adaptado de (Zhai y Massung, 2016)

La palabra **persona** de la figura indica el titular o dueña de la opinión. Cuando se habla de opinión, es sobre una opinión sostenida por **alguien**, por supuesto, que la opinión dependerá de la cultura, antecedentes, y contexto en general. Este proceso muestra que hay múltiples elementos que se requiere incluir a fin de caracterizar una opinión.

Una opinión está representada básicamente por tres elementos. Primero, tiene que especificar quien es el titular de la opinión. Segundo, especifica el objetivo, o de qué es la opinión. Tercero, el contenido de la opinión propiamente. Si se identifican estos elementos, entonces se comprende una opinión. Para profundizar su entendimiento se puede además identificar dos elementos más: el contexto de la opinión y la situación en la que la opinión fue expresada. Además de comprender el sentimiento de la opinión, si es positiva o negativa.

Tarea de la minería de opinión

La tarea de la minería de opinión puede ser definida como tomar entradas contextualizadas para generar un conjunto de representaciones de opinión, como lo muestra la Figura 15. Cada representación debería identificar al titular de la opinión, objetivo, contenido, y contexto. Idealmente, se inferirá el sentimiento de la opinión de un comentario y el contexto para comprender mejor la opinión.

La minería de opinión es importante y útil por las siguientes razones:

- Se aplica para apoyar en la toma de decisiones, ya que a menudo se considera la opinión de otras personas al leer sus comentarios para tomar una decisión sobre que producto comprar, o que servicio usar. También estaría interesado en la opinión de otros para decidir por quien votar. Incluso los políticos están interesados en conocer la opinión de sus electores cuando diseñan nuevas políticas.
- Se aplica para comprender a las personas. Por ejemplo, comprender las preferencias de los humanos, puede optimizar la búsqueda de un producto u optimizar un sistema de recomendación si se conoce en que están interesadas las personas. Además ayuda en el proceso de anuncios, se puede tener anuncios direccionados si se conoce los gustos y preferencias de ciertos tipos de personas con ciertos tipos de productos.
- Se aplica para resumir un conjunto de opiniones de muchas personas en una sola para valorar una opinión más general. Es muy útil en la inteligencia de negocios así las manufactureras conocerán donde sus productos tienen ventajas y desventajas. ¿Cuáles son las características ganadoras de sus productos o de sus competidores? La investigación de marketing se realiza con el entendimiento de las opiniones de los consumidores.

La investigación de la ciencia social manejada por datos puede beneficiarse realizando minería de texto para comprender las opiniones de grupos. Y si se adiciona las opiniones de los medios sociales, se puede estudiar el comportamiento de las personas en redes sociales.

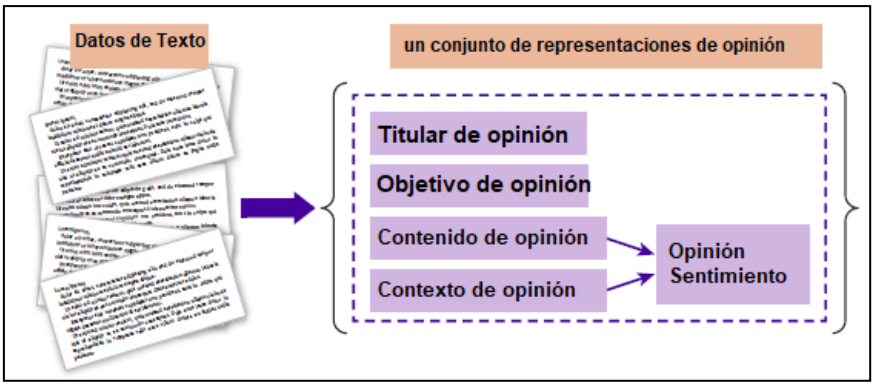


Figura 15. La tarea de la minería de opinión

Fuente: Adaptado de (Zhai y Massung, 2016)

1.1.5. Proceso de análisis de opinión

El análisis de opinión o clasificación sentimental (Zhai y Massung, 2016) puede ser definida como: la entrada es un objeto de texto opinado y la salida es típicamente una etiqueta sentimental que puede ser definido de dos formas. Uno es el **análisis de polaridad**, donde se tienen categorías tales como positivo, negativo, o neutral, Figura 16. El otro es el **análisis emocional** que puede ir más allá de la polaridad para caracterizar la sensación precisa del titular de la opinión. En el caso del análisis de la polaridad, a veces se tiene puntuaciones numéricas como se ven en las revisiones de la Web. Una puntuación de cinco podría denotarlo como lo más positivo, y uno podría ser el más negativo, por ejemplo. En el análisis emocional hay también diferentes formas de diseñar las categorías. Algunas categorías son feliz, triste, temor, molesto, sorpresa, y disgusto. Así la tarea es esencialmente una tarea de clasificación, o tarea de categorización.

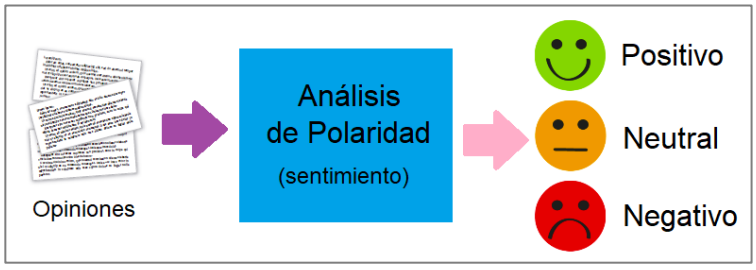


Figura 16. Análisis de la Polaridad o sentimiento

Si simplemente se aplica las técnicas por defecto de clasificación, la precisión no podría ser buena ya que la clasificación sentimental requiere algunas mejoras sobre las técnicas regulares de categorización de texto. En particular, se necesita dos tipos de mejoras. Uno es usar características más sofisticadas que puedan ser más

apropiadas para el análisis sentimental. Y la otra es considerar el orden de las categorías, especialmente en el análisis de polaridad ya que hay claro orden entre las elecciones. Por ejemplo, se puede usar **regresión logística** para predecir el valor dentro de algún rango.

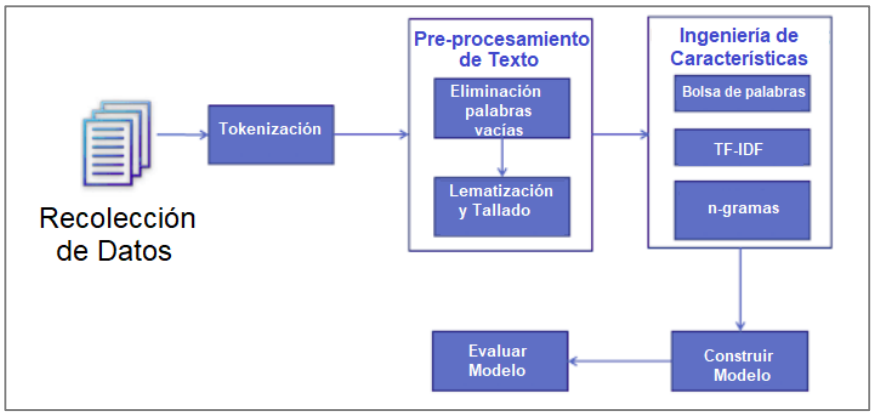


Figura 17. Proceso del análisis de opinión o clasificación sentimental

Fuente: Adaptado de (Liu, 2012)

El proceso de clasificación de sentimiento o análisis de opinión sigue los pasos mostrados en la Figura 17.

A. Recolección de Datos y construcción de Dataset

Se trata de la reunión de datos de texto necesario para el análisis de texto, esta tarea es importante porque es usado como muestra de entrenamiento para construir ya sea modelos de clasificación o extracción de texto.

En el análisis de opinión, se requiere reunir comentarios que se encuentren internamente o externamente; los comentarios de nivel interno se pueden provenir de correos electrónicos, chats, comentarios de los clientes, consultas personalizadas, y tickets de atención del cliente, así mismo a nivel externo se puede recolectar comentarios desde diferentes sitios web, para ello se utiliza herramientas de *crawling* y APIs que permiten obtener datos desde sus plataformas, por ejemplo Facebook, Twitter e Instagram. Adicionalmente existen datos abiertos o conjuntos de comentarios que están disponibles en sitios web como Kaggle y Quandl.

B. Pre procesamiento de Texto

La preparación de datos se realiza para construir la entrada de los procesos de aprendizaje de máquina para realizar el análisis de datos. Se hace uso de las técnicas de Procesamiento de Lenguaje Natural (PLN).

a. Tokenización

Para reconocer las unidades por analizar se debe tokenizar el texto, esta tarea se ocupa de cortar una cadena de caracteres (texto) en partes semánticamente significativa que puede ser analizadas (p.ej. palabras) descartando trozos sin sentido (p.ej. espacios en blanco).

b. Eliminación de palabras sin utilidad

Para proveer un análisis automatizado más preciso del texto, es importante eliminar aquellas palabras que son muy frecuentes y que no proporcionan información, a este tipo de palabras se les conoce como *palabras vacías*, estas listas de palabras son diferentes en cada lenguaje, entonces esta tarea se realiza dependiendo del texto que se va analizar y el análisis que se desea realizar. Además se realiza algún análisis léxico desde el dominio del texto de donde proviene a fin de determinar las palabras que deberían ser agregadas a la lista de palabras vacías.

Dependiendo del problema en mano, las secuencias de números, URLs y algunos nombres no son relevantes para la detección de una opinión, entonces estas palabras también se deben agregar a la lista de palabras vacías.

c. Lematización y Tallado

Esta tarea se refiere al proceso de remover todos los afijos colgados en una palabra para mantener su base léxica, se le conoce como raíz o tallo o su forma de diccionario o lema. La diferencia entre estas tareas se basan en las reglas para cortar los principios o finales de una palabra, así la lematización hace uso de diccionarios y de análisis morfológicos más complejos.

C. Ingeniería de características, construir el vocabulario y generar vectores.

Los algoritmos de análisis de texto requieren que el texto tenga una forma numérica, para ello es necesario convertir el texto en vectores numéricos; entonces se crea un vocabulario y este vocabulario será calificado o puntuado. Los pasos se muestran en la Figura 18.

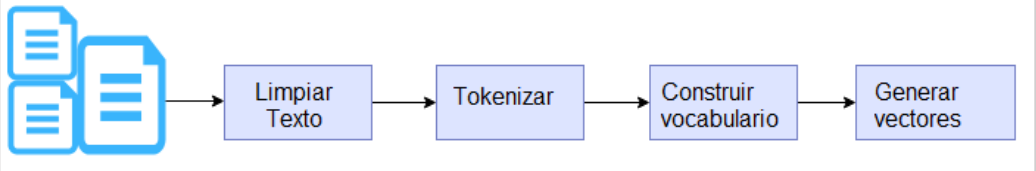


Figura 18. Pasos de la construcción de características y generación de vectores

Fuente: Adaptado de (Jurafsky y Manning, 2012)

a. Bolsa de palabras

Es una técnica de extracción de características o codificación de características de texto, en el que se representa los datos de texto en un conjunto de palabras en unidades básicas y se ignora el orden y estructura de las mismas para aplicar los algoritmos de aprendizaje de máquina. Se aplica debido a que el texto es confuso y los algoritmos de aprendizaje de maquina requieren de entradas muy bien definidas y de tamaño fijo, ya que los algoritmos de aprendizaje de máquina no pueden trabajar con texto sin estructura directamente, por lo que el texto debe convertirse en números, específicamente en **vectores de números**. El modelo tiene que ver sólo con las palabras conocidas dentro del documento de texto y no en qué parte del documento se encuentran. Figura 19-a.

b. N-gramas

Es una solución más sofisticada de crear un vocabulario en grupos de palabras. La agrupación de palabras amplia el ámbito del vocabulario y permite capturar un poco más significativo de los documentos. En esta aproximación cada palabra o token se llama una "grama". Si se crea un vocabulario de dos-palabras se llama bi-grama. Un N-grama es una secuencia de n-tokens de palabras para un vocabulario de n-gramas o modelo de n-gramas donde n se refiere a n palabras agrupadas. Figura 19.b.

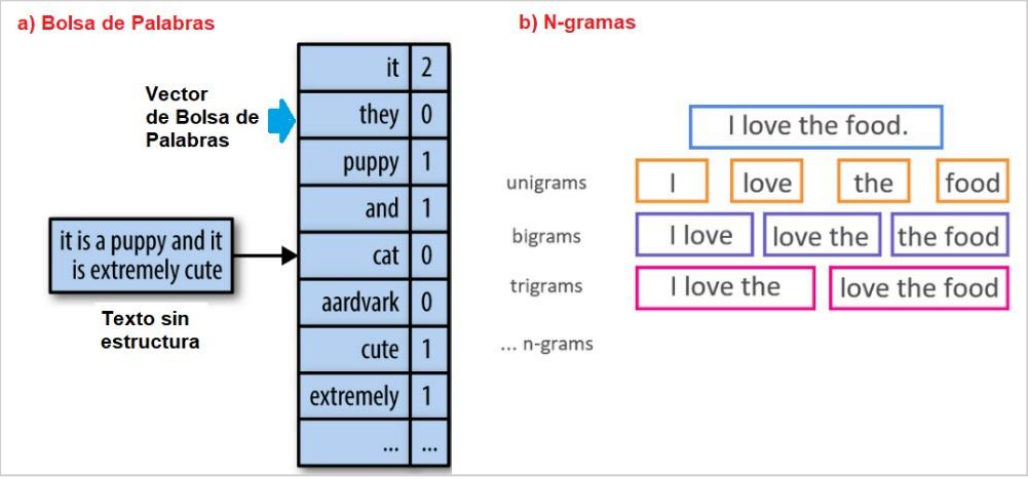


Figura 19. Bolsa de palabras y n-gramas

Una vez que el vocabulario ha sido seleccionado, la ocurrencia de palabras en los documentos deben ser calificados, los métodos que se aplican son: conteo y frecuencias. El conteo de palabras es la cuenta de las veces que cada palabra

aparece en un documento, mientras que las frecuencias calculan la frecuencia que cada palabra aparece en un documento fuera de todas las palabras en el documento.

a. Hashing de palabras

Una función hash es una función matemática que mapea los datos a un tamaño fijo de conjunto de números. Se usa una representación hash para cada palabra conocida en el vocabulario. A las palabras se les aplica la función hash determinísticamente al mismo índice entero en el espacio hash objetivo. Una puntuación binaria o conteo puede luego ser usado para puntuar la palabra. El desafío es elegir un espacio hash para acomodar el tamaño del vocabulario elegido para minimizar la probabilidad de colisiones y el intercambio de esparcimiento.

b. TF-IDF

La frecuencia de términos (TF) es la puntuación de la frecuencia de palabras en el documento reciente y la frecuencia inversa de documentos (IDF) es la puntuación de cuan raro es la palabra a través de los documentos. TF-IDF es la solución a la puntuación de la frecuencia de palabras es alta en aquellas palabras que dominan en el documento pero que no contiene tanta información para el modelo como palabras más raras pero de dominio específico, el cálculo de TF-IDF se muestra en la Figura 20.

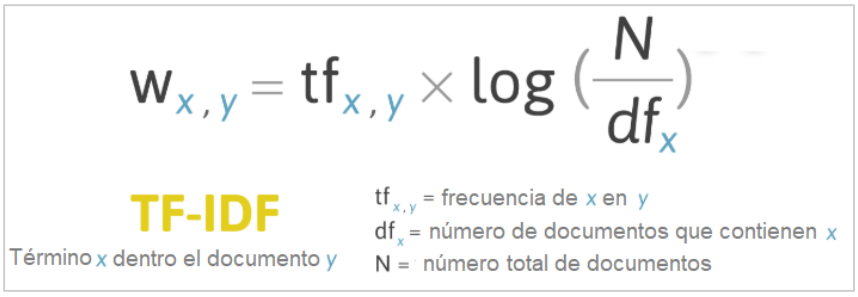


Figura 20. Calculo de la Frecuencia de Términos

D. Construir el modelo de análisis de datos

La clasificación de texto o categorización de texto o etiquetado es el proceso de asignar etiquetas al texto basado en su contenido. El análisis de texto incluye el análisis de sentimiento (p.ej. detectar cuando un texto dice algo positivo o negativo sobre un tema), detección de temas (p.ej. determinar de qué temas un texto habla),

y detección de intensión (p.ej. detectar el propósito o intensión subyacente del texto).

Los sistemas basados en reglas en la clasificación de texto, una regla es esencialmente una asociación hecha por humanos entre el patrón lingüístico que puede ser hallado en un texto y una etiqueta. Los que los sistemas basados en reglas hacen es detectar estor patrones lingüísticos hechos a mano en los textos y asignar las etiquetas correspondientes basadas en los resultados de la detección, normalmente las reglas consisten de referencias morfológicas, léxicas, o patrones sintácticos, incluyendo elementos semánticos y morfológicos. Sin embargo los modelos basados en reglas toman mucho tiempo y requieren del conocimiento de tanto la lingüística y el tema que se esté tratando en los textos que se está analizando por lo que son difíciles de escalar y mantener.

Los sistemas basados en aprendizaje de máquina, pueden hacer predicciones basados en lo que aprenden a partir de observaciones pasadas. Estos sistemas necesitan ser alimentados con muchos ejemplos de textos y las predicciones esperadas (etiquetas) de cada uno de ellos. Mientras mejores sean las muestras que alimenten el clasificador, mejor será la predicción. Estas muestras se llaman datos de entrenamiento, que deben ser transformados en vectores (vectorización de la bolsa de palabras) a partir del cual el sistema de aprendizaje de máquina extraerá características relevantes que lo ayudaran a aprender a partir de los datos existentes y hacer predicciones sobre los textos que vengan.

E. Evaluar el modelo

El funcionamiento de un clasificador es usualmente evaluado a través de métricas estándares usadas en el campo de aprendizaje de máquina. Estas métricas son: exactitud (*accuracy*), precisión (*precision*), recuperación (*recall*) y F1. Permitiendo comprender cuan bueno es el clasificador en el análisis de texto. Además la evaluación se puede realizar en conjuntos de prueba fijos (p.ej. un conjunto de textos del que se conoce las etiquetas resultantes) o usar evaluación cruzada (p.ej. un método que divide los datos de entrenamiento en diferentes pliegues de tal forma que puedan ser usados algunos subconjuntos de los datos para el entrenamiento y otros para la prueba).

1.1.6. Aprendizaje supervisado para la clasificación de opinión

Los algoritmos de aprendizaje de máquina que permiten la clasificación de opinión son de aprendizaje supervisado (Zafarani *et al.*, 2014). Son aquellas en que los valores de los atributos de clase del *dataset* son conocidas antes de ejecutar el algoritmo. Este dato se llama *dato etiquetado* o datos de *entrenamiento*. Las instancias en este conjunto son tuplas de la forma (**x**, y), donde **x** es un vector y **y** es el atributo de clase, comúnmente un escalar. El aprendizaje supervisado construye un modelo que mapea **x** a **y**. Aproximadamente, la tarea es hallar un mapeamiento m(.) tal que m(x)=y. Luego el conjunto de datos sin etiqueta o *dataset* de prueba, en que las instancias están en la forma (x,?) y los valores de **y** son desconocidos. Dado m(.) aprendido desde los datos de entrenamiento y **x** de una instancia sin etiqueta, se computa m(x), el resultado de la predicción de la etiqueta para la instancia sin etiqueta. Los métodos de clasificación son: aprendizaje de árbol de decisiones, clasificador naive Bayes, clasificador del vecino más cercano k-nearest, y clasificación con información de red; además los métodos de regresión tales como regresión lineal y regresión logística.

Regresión Logística

Es un método que permite predecir una respuesta binaria. Es un caso especial de modelos lineales generalizados que predicen la probabilidad de un resultado. La regresión logística mide la relación entre "Etiqueta" Y y las "Características" X estimando las probabilidades usando una **función logística**, el modelo predice una probabilidad que es usado para predecir la clase de etiqueta.

La clasificación de texto usa la regresión logística para predecir la probabilidad de un comentario de texto sea positivo o negativo, dada la etiqueta y el vector característico de los valores TF-IDF. La regresión logística encuentra el mejor **peso** que encaje en cada palabra en la colección del texto multiplicando cada característica TF-IDF por un peso y pasando la suma a través de una función sigmoidea (forma S), que transforma la entrada X en una salida Y, entre un número de 0 y 1. En otras palabras la regresión logística puede ser entendida como encontrar los **parámetros que mejor calcen**. La regresión logística tiene las siguientes ventajas: puede manejar el esparcimiento de los datos; es rápido de entrenar; los pesos pueden ser interpretados, los pesos positivos corresponderán a las palabras

que son positivas, y los pesos negativos corresponderán a las palabras que son negativas.

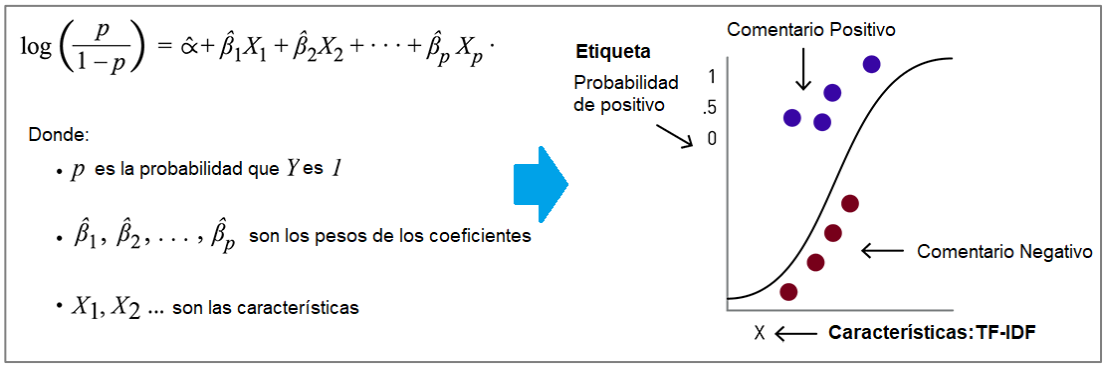


Figura 21. Regresión Logística

Fuente: Adaptado de (Singh, 2018)

1.1.7. Evaluación del análisis de opinión de aprendizaje supervisado

Para evaluar un modelo de clasificación se usa las métricas estándares del aprendizaje de máquina para estas tareas: exactitud o *accuracy*, precisión, recuperación o *recall* y medida F. La Tabla 1 muestra las fórmulas de cálculo de cada una de ellas.

La precisión a un valor conocido es el número de predicciones correctas que el clasificador ha realizado dividido entre el número de predicciones. La precisión a medidas cercanas declara cuantos textos fueron pronosticados correctamente fuera de unos que fueron pronosticados como pertenecientes a una etiqueta dada. Así el número de textos que fueron correctamente pronosticados como positivos para una etiqueta dada y la divide por el número de textos que fueron pronosticados (correctos e incorrectos) como pertenecientes a la etiqueta.

La recuperación declara cuantos textos fueron pronosticados correctamente fuera de los que deberían haber sido pronosticados como pertenecientes a una etiqueta dada. Entonces se toma el número de textos que fueron correctamente pronosticados como positivos para una etiqueta dad y las divide por el número de textos que ya fueron pronosticados correctamente como pertenecientes a la etiqueta o que fueron incorrectamente pronosticados como no pertenecientes a la etiqueta.

El puntaje F1 es la media armónica de la precisión al valor conocido y la recuperación. Dice cuan bien el clasificador funciona y si es de igual importancia

es la precisión de medidas y de recuperación. Así, el puntaje F1 es un indicador mucho mejor del funcionamiento del clasificador que la precisión del valor conocido.

La validación cruzada se usa a menudo para evaluar el funcionamiento del clasificador de texto. El método consiste en dividir aleatoriamente el conjunto de datos de entrenamiento en subconjuntos de igual tamaño (p.ej. 4 subconjuntos con 25% de los datos originales cada uno). Luego, todos los subconjuntos excepto uno que es usado para entrenar el clasificador (en este caso 3 subconjuntos con 75% de los datos originales) y este clasificador es usado para predecir los textos en los subconjuntos restantes.

Luego se usa las métricas como la precisión por valor conocido, precisión por medidas, recuperación y F1, finalmente el proceso es repetido con un nuevo pliegue de prueba hasta que todos los pliegues hayan sido usados, la métrica del promedio del funcionamiento se computa y el proceso de evaluación se termina.

Tabla 1

Calculo de las medidas de evaluación de modelos de clasificación

Medida	Formula
Precisión	$Precisión = \frac{VP}{VP + FP}$
Recuperación / Sensibilidad	$Recuperación\ o\ Sensibilidad = \frac{VP}{VP + FN}$
Selectividad	$Selectividad = \frac{VN}{FP + VN}$
Exactitud	$Exactitud = \frac{VP + VN}{VP + VN + FP + FN}$
Medida-F	$Medida\ F = \frac{2 * precisión * Recuperación}{Precisión + Recuperación}$

Fuente: Adaptado de (Zhai y Massung, 2016)

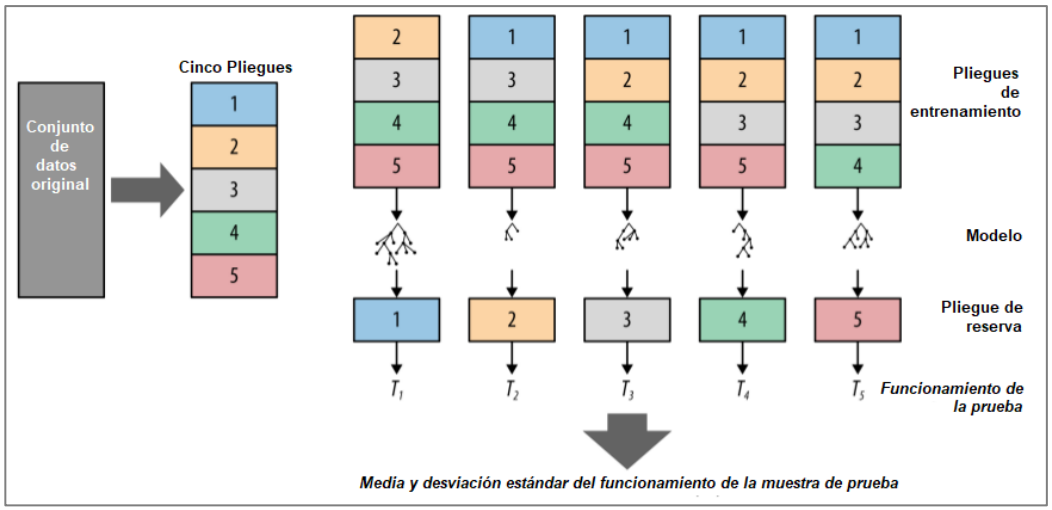


Figura 22. Validación cruzada

Fuente: Adaptado de (Lopez Briega, 2016)

1.1.8. El framework Spark

Spark es un *framework* que manipula conjuntos de datos masivos con procesamiento paralelo y alta velocidad usando mecanismos robustos, como el Big Data requiere de procesamientos de datos escalables y rápidos, Spark permite que se realicen en forma paralela y distribuida la computación de procesamiento de datos paralelizando las tareas y acumulando los resultados al final (Singh, 2018).

Apache Spark comenzó como un proyecto de investigación en el laboratorio AMPLab de la UC Berkeley en el 2009 y fue se abrió su código en 2010, desde entonces ha ido evolucionando hasta hoy como lo muestra la Figura 23.

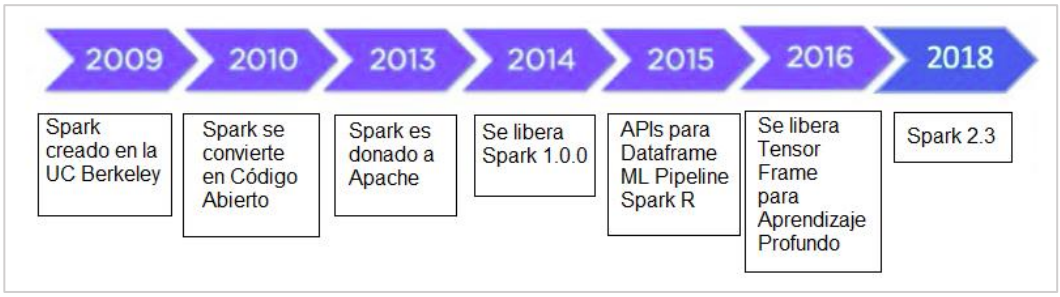


Figura 23. Evolución de Spark

Fuente: Adaptado de (Singh, 2018)

Apache Spark al ser un sistema computacional unificado más un conjunto de librerías o APIs para el procesamiento paralelo en *clusters* de computadoras. Spark es el sistema más activamente desarrollado de código abierto para este tipo de tareas, convirtiéndose en la herramienta estándar para cualquier desarrollador o

científico de datos interesado en Big Data. Spark soporta múltiples lenguajes de programación ampliamente usados (Python, Java, Scala, y R) para desarrollar una aplicación de procesamiento de datos, e incluye librerías para diversas tareas que van desde SQL a *streaming* y aprendizaje de máquina, y se ejecuta en cualquier sitio desde una laptop hasta un *cluster* de miles de servidores. Así, Spark es un sistema sencillo para comenzar y proporcionalmente escalar hacia el procesamiento real de Big Data(Chambers y Zaharia, 2018). La Figura 24 muestra el contexto de Spark.

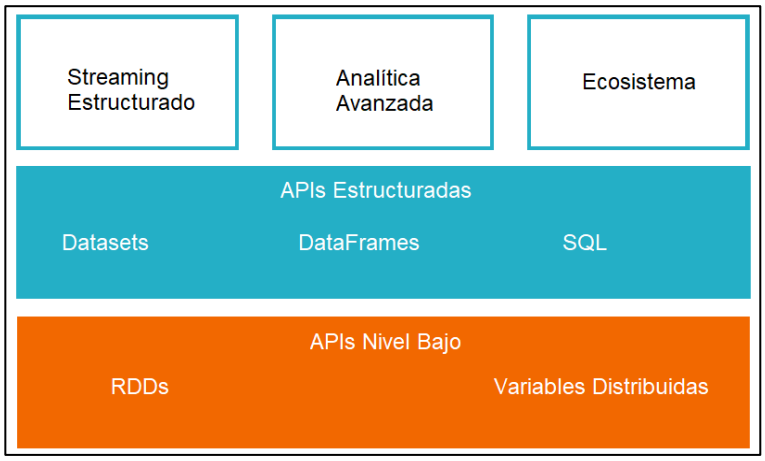


Figura 24. Caja de herramientas de Spark

Fuente: Adaptado de (Chambers y Zaharia, 2018)

1.1.8.1. La Filosofía Spark

Spark fue diseñado en función a tres fundamentos: unificado, sistema computacional y librerías.

Unificado, Spark ofrece un sistema unificado para escribir aplicaciones de Big Data; diseñado para soportar un amplio rango de tareas de analítica de datos, desde una simple carga de datos y consultas SQL a computación de *streaming* y aprendizaje de máquina sobre un mismo sistema computacional y un consistente conjunto de librerías.

Antes de Spark, ningún sistema de código abierto suministraba un sistema unificado para el procesamiento de datos en paralelo de código abierto, los usuarios tenían que unir varias aplicaciones de múltiples APIs y sistemas, ya que las tareas de analítica de datos del mundo real requieren de la combinación de diferentes tipos de procesamiento y librerías.

Sistema computacional, Spark limita cuidadosamente su alcance a sólo un sistema computacional, Spark solo maneja las cargas de datos desde los sistemas de almacenamiento y realiza computaciones sobre ellos, no es un sistema de almacenamiento sino de las computaciones sobre los datos. Se puede usar Spark junto a sistemas de almacenamiento persistente, y sistemas de almacenamiento en la nube como Azure Storage y Amazon S3, sistemas de archivos distribuidos como Apache Hadoop, almacenes *key-value* como Apache Cassandra, y buses de mensajes como Apache Kafka. Esta característica hace diferente a Spark de las primeras plataformas de software para Big Data como Apache Hadoop que incluye tanto el sistema de almacenamiento y el sistema computacional (MapReduce) en uno solo.

Librerías, las librerías de Spark están diseñadas como parte del sistema unificado que suministra una API unificada en común para las tareas de análisis de datos. Spark soporta tanto las librerías estándar internas como las externas de otras comunidades de código abierto. Actualmente las librerías estándar son la mayor parte del proyecto de código abierto, ya que el sistema principal solo ha tenido algunos cambios desde su lanzamiento. Las librerías han ido creciendo para suministrar más tipos de funcionalidad, como librerías para SQL y datos estructurados (Spark SQL), aprendizaje de máquina (MLlib), procesamiento de *streaming* y analítica de grafos (GraphX).

1.1.8.2.Arquitectura básica de Spark

Actualmente el procesamiento de datos es un área particular de mucho desafío, ya que una sola computadora no tiene el poder y recursos suficientes para realizar computaciones con grandes cantidades de información. Un *cluster*, o grupo de computadoras, junta recursos de muchas máquinas para tener la habilidad de usar todos los recursos acumulados como si fueran una sola. Y para ello es necesario un *framework* que coordine el trabajo a través de ellos. Por lo tanto Spark administra y coordina la ejecución de tareas en datos a través de un *cluster* de computadoras.

El *cluster* de máquinas que Spark usa para ejecutar tareas es administrado por un administrador de *cluster* como un administrador de *cluster* autónomo, YARN o Mesos. Y luego se enviaran las aplicaciones Spark a los administradores de

cluster, que concederán recursos a las aplicaciones para que completen sus trabajos.

Aplicaciones Spark

Las aplicaciones de Spark consisten de un proceso *driver* o conductor y un conjunto de procesos ejecutores. El proceso conductor ejecuta la función main(), se coloca en un nodo en el *cluster*, y es responsable de tres cosas: mantener la información sobre la Aplicación Spark; responder a un programa de usuario o entrada; y analizar, distribuir, y planificar el trabajo a través de los ejecutores. El proceso conductor es absolutamente esencial, es el corazón de una Aplicación Spark y mantiene toda la información relevante durante el tiempo de vida en la aplicación.

Los procesos ejecutores son responsables por actualmente realizar el trabajo que el conductor les asigna. Eso significa que cada ejecutor es responsable de únicamente dos cosas: ejecutar el código asignado por el conductor, y reportar el estado de la computación sobre lo que el ejecutor retorna al nodo conductor. La Figura 25 demuestra como el administrador del *cluster* controla físicamente las máquinas y asigna recursos a las Aplicaciones Spark. Este puede ser uno de tres administradores principales del *cluster*, administrador de *cluster* autónomo de Spark, YARN o Mesos. Lo que significa que pueden estar múltiples aplicaciones Spark ejecutándose en un *cluster* al mismo tiempo.

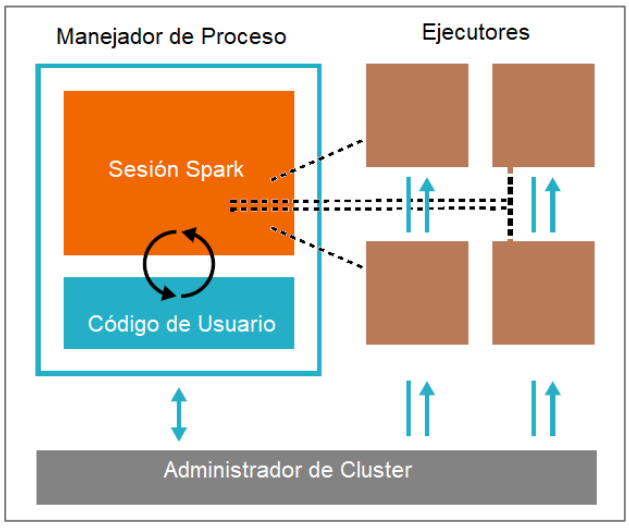


Figura 25. Arquitectura de una aplicación Spark

Fuente: Adaptado de (Chambers y Zaharia, 2018)

Spark, adicionalmente de su modo en cluster, también tiene un modo local. El conductor y los ejecutores son simplemente procesos, lo que significa que pueden vivir en la misma máquina o diferentes máquinas. En el modo local, el conductor y los ejecutores se ejecutan como hilos en la computadora individual en lugar de un *cluster*.

1.1.8.3.Lenguaje de las librerías de Spark

Los lenguajes de las librerías de Spark hacen posible que se pueda ejecutar código Spark usando varios lenguajes de programación. La mayor parte, Spark presenta algunos conceptos principales en cada lenguaje; estos conceptos son luego traducidos a código Spark que se ejecuta en el cluster de computadoras. La Figura 26 muestra la relación de los lenguajes de programación con el *framework* Spark.

Scala, Spark ha sido escrito principalmente en Scala, convirtiéndola en el lenguaje por defecto.

Java, aunque Spark ha sido escrito en Scala, los autores de Spark han sido cuidadosos de asegurar que se pueda escribir código también en Java.

Python, soporta casi todos los constructores que Scala soporta.
SQL, Spark soporta un subconjunto del estándar ANSI SQL 2003. Haciendo fácil a los analistas y no programadores tomar ventaja de los poderes de Big Data de Spark.

R, Spark tiene dos librerías comúnmente usadas de R: una como parte del núcleo de Spark (SparkR) y otro como un paquete de Re manejada por la comunidad.

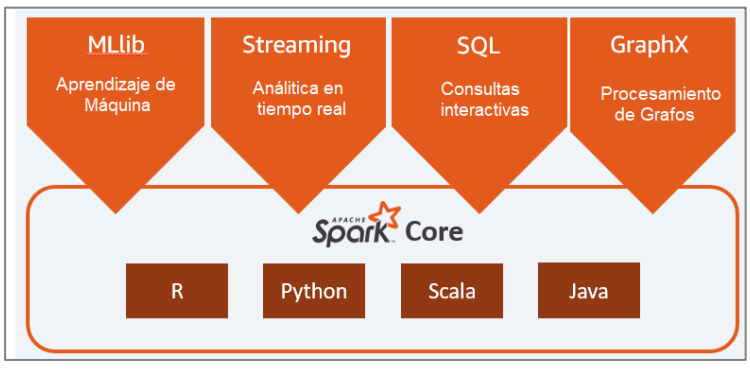


Figura 26. Lenguajes de programación de las librerías de Spark
Fuente: Adaptado de (Zaharia *et al.*, 2016)

Cada librería de cada uno de los lenguajes soportado por Spark mantiene los mismos conceptos básicos descritos anteriormente. Hay un objeto **SparkSession** disponible al usuario, que es el punto de entrada para ejecutar el código Spark. Cuando se usa Spark desde Python o R, no se escribe explícitamente instrucciones JVM; en su lugar, se escribe código en Python y R que Spark traducirá en código que pueda ser ejecutado por los ejecutores JVMs. Esta relación se muestra en la Figura 27.

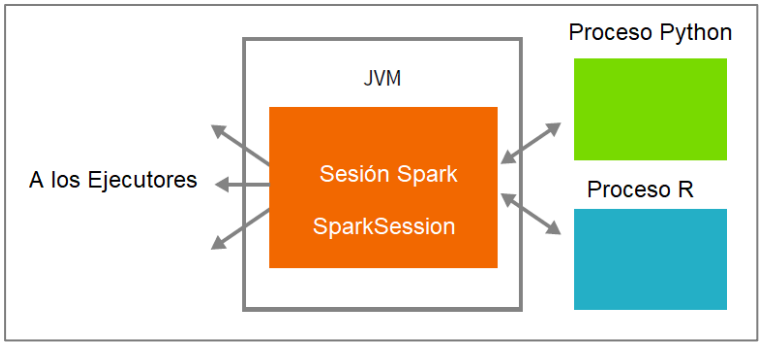


Figura 27. La Relación entre SparkSession y los Lenguajes de la API Spark
Fuente: Adaptado de (Chambers y Zaharia, 2018)

1.1.8.4.Estructura de datos de Spark

Las librerías "no estructuradas" o de bajo-nivel usan RDD, y las librerías estructuradas de alto-nivel usan los DataSet y DataFrame para manipular todos los tipos de datos y poder distribuirlos en el *cluster* y se realicen las operaciones respectivas.

RDD, Resiliente Distributed Dataset representa una colección de registros inmutables, particionado que pueden ser operados en paralelo. Los RDD son objetos Java o Python de elección del programador, que dan un control completo para almacenar lo que sea, en cualquier formato que se desee dentro de estos objetos. Cada iteración y manipulación entre los valores tienen que ser definidos explícitamente por el programador, o "reinventar la rueda" para cualquier tarea que se trate de realizar en Spark, incluso las optimizaciones. Al usar un RDD, Spark deshabilita sus funciones automáticas de planificación y administración de recursos para la ejecución de una aplicación.

DataSets, son los tipos fundamentales de la API estructurada, tienen la característica del lenguaje estrictamente de Java Virtual Machine que funciona únicamente con Scala y Java, permite definir el objeto que cada fila del que consistirá el DataSet. En Scala, es el caso de un objeto de clase que esencialmente define un esquema que se puede usar, y en Java es definir un Java Bean. Se usará un Dataset cuando la operación que se realice no pueda ser expresado usando manipulaciones con un DataFrame, y cuando se quiera o necesite un tipo seguro aunque implique un menor rendimiento en la ejecución de la aplicación.

DataFrames, es la librería estructurada común y representa simplemente una tabla de datos con filas y columnas, como una hoja de cálculo con nombre de columnas. Los DataFrames se acompañan de un esquema, compuesto por una lista que define las columnas y los tipos dentro de cada columna. Un DataFrame se esparce en miles de computadoras por defecto así se puede realizar computaciones en datos muy grandes que no se ajustan a la memoria de una sola computadora o que puede llevar mucho tiempo computarla en una sola.

El concepto de DataFrame no es único solo en el mundo de Spark, tanto R y Python tienen conceptos similares. Sin embargo, los DataFrames de R y Python funcionan en una sola computadora y no en múltiples. Lo que limita que se puede hacer con un DataFrame de Python y R con los recursos de una sola computadora a la de Spark con muchas (cluster).

Ya que Spark tiene tres conjuntos fundamentales de APIs para abstraer y organizar los datos: Datasets, DataFrames, SQL Tables, y Resilient Distributed Datasets (RDD). El más fácil y eficiente de utilizar es el DataFrame, que está disponible en todos los lenguajes que soporta Spark. Todas estas abstracciones representan colecciones distribuidas de datos pero que tienen diferentes interfaces para trabajar con ellos, tanto los Datasets y los DataFrames forma parte de las mejoras del *framework* Spark desde la versión 2.0.

1.1.8.5.Procesamiento de datos con Spark

Los siguientes conceptos son propios del ecosistema de Spark, y que permiten comprender su modo de trabajo para procesar datos.

a. **Particiones,** para que cada ejecutor funcione en paralelo, Spark divide los datos en pedazos (*chunks*) o particiones. Cada partición es una colección de filas colocadas en una máquina física del *cluster*. Por ejemplo, las particiones de DataFrames representan como los datos son físicamente distribuidos a través del cluster durante la ejecución de la aplicación y que no son manipulados manualmente, ya que sólo se especifica las transformaciones de alto nivel de datos en las particiones físicas y Spark determina como el trabajo se ejecutará dentro del cluster.

b. **Transformaciones,** en Spark una transformación es una instrucción que permite modificar los datos, las estructuras de datos principales en Spark son *inmutables* lo que significa que no pueden ser modificados luego que hayan sido creadas. Las transformaciones suministran las bases para construir la lógica de negocio de la aplicación, pudiéndose usar las transformaciones de corta o amplia dependencia. Cuando una transformación es de reducida dependencia significa que cada partición de entrada contribuirá en una sola partición de salida, mientras que una transformación de amplia dependencia contribuirá a muchas particiones de salida. Analógicamente a la idea de MapReduce, Map es a una transformación de reducida dependencia como Reduce es a una transformación de amplia dependencia. Figura 28.

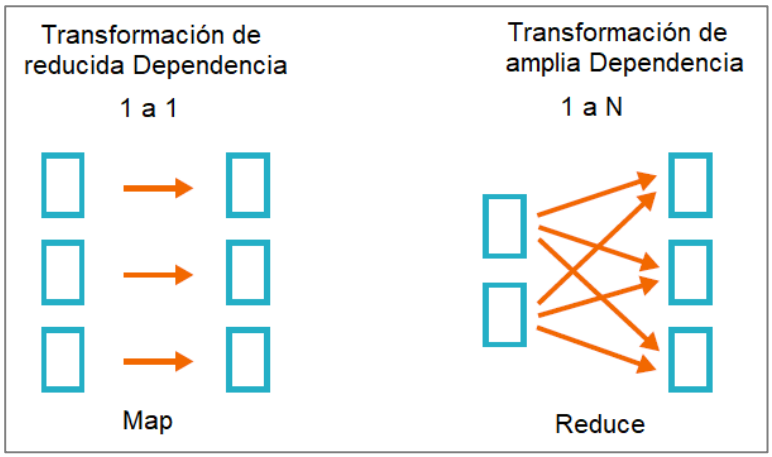


Figura 28. Transformaciones Spark analógicas a MapReduce

Fuente: Adaptado de (Chambers y Zaharia, 2018)

c. **Operaciones Ociosas,** Spark espera hasta el último momento para ejecutar el flujo de las instrucciones computacionales. En vez de modificar los datos inmediatamente cuando se expresa alguna operación, se construye un plan de transformaciones que se va aplicar a los datos. Entonces, Spark compilara

el plan desde los datos sin estructura, transformaciones DataFrame, a un plan físicamente eficiente que se ejecutara los más eficientemente posible a través del *cluster*. Así Spark optimiza el flujo de datos por completo de principio a fin.

d. **Acciones,** las transformaciones permiten construir el plan lógico de transformación, para ejecutar las computaciones, se ejecutara una acción. Una acción instruye a Spark a computar el resultado desde una serie de transformaciones. Existen tres clases de acciones: ver los datos en consola, recolectar datos a objetos nativos en el lenguaje respectivo, escribir la salida de los recursos de datos.

1.1.8.6.Aprendizaje de máquina y analítica avanzada en Spark

Un aspecto popular de Spark es su habilidad de realizar aprendizaje de máquina de gran escala con una biblioteca incorporada de aprendizaje de máquina llamada MLlib, y Spark ML. Spark ML introducido desde la versión 2.0 permite el pre-procesamiento, transformación de datos, entrenamiento de modelos, y realización de predicciones sobre datos en escala. Spark suministra una API de aprendizaje de máquina sofisticada para realizar una variedad de tareas de aprendizaje de máquina, desde clasificación a regresión, agrupamiento a *deep learning*.

1.1.8.6.1. Spark ML Pipelines

Spark ML (Apache Spark, 2019) suministra un conjunto uniforme de APIs de alto nivel construido sobre DataFrames con el objetivo de ayudar a crear y afinar la cola de lógica de datos (AI Zone) de aprendizaje de máquina práctico. MLlib estandariza las APIs para los algoritmos de aprendizaje de máquina para combinar los múltiples algoritmos dentro de un simple flujo de trabajo.

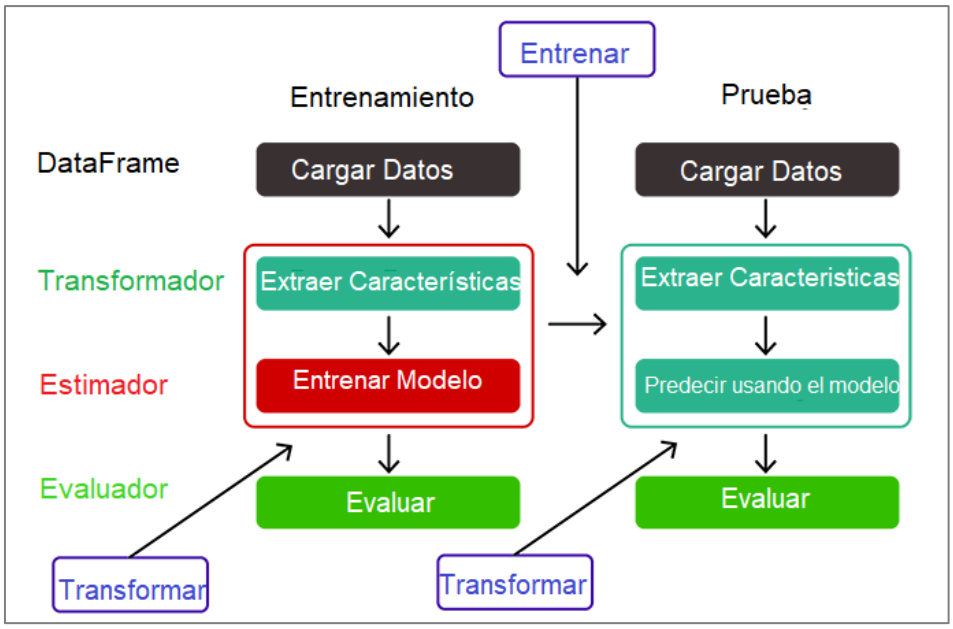


Figura 29. Flujo de Trabajo de Spark MLlib

Fuente: Adaptado de (AI Zone, 2019)

Los componentes del flujo de trabajo mostrados en la Figura 29 se describen como sigue:

- **DataFrame**, la API ML use DataFrame de Spark SQL como un *dataset* ML, que puede mantener una variedad de tipos de datos, p.ej. un DataFrame podría tener diferentes columnas almacenando texto, vectores característicos, etiquetas verdaderas, y predicciones.
- **Transformer**, un transformador es un algoritmos que puede transformar un DataFrame en otro DataFrame, p.ej. un modelo ML es un Transformer que transforma un DataFrame con características en un DataFrame con predicciones. Técnicamente implementa un método transform().
- **Estimator**, un estimador es un algoritmo que puede ser entrenado en un DataFrame para producir un Transformer, p.ej. un algoritmo de aprendizaje es un estimador que entrena sobre un DataFrame y produce un modelo. Técnicamente un estimador implementa un método fit().
- **Parameter**, todos los transformadores y estimadores comparten una API en común para parámetros específicos.
- **Pipeline**, encadena múltiples transformadores y estimadores junto a un flujo de trabajo ML especifico. En aprendizaje de máquina es común de ejecutar una secuencia de algoritmos para procesar y aprender a partir

de los datos, este flujo de trabajo consiste de una secuencia de *PipelineStageS* (transformadores y estimadores) que se ejecutan en un orden específico conformando un conjunto de etapas. Las etapas de un estimador para un flujo de trabajo de documentos de texto y usado para entrenar un modelo de aprendizaje de máquina se muestra en la Figura 30, donde el Pipeline se conforma de tres etapas, los primeros dos son transformadores (Tokenizador y HashingTF) y un tercero es el estimador (RegresiónLogistica) produciendo un PipelineModel que es un transformador usado para probar el modelo.

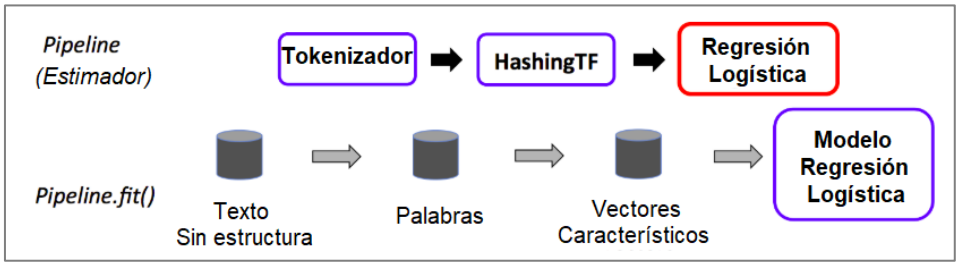


Figura 30. Pipeline de Entrenamiento del Modelo ML

Fuente: Adaptado de (Apache Spark, 2019)

Una vez que se ha obtenido el PipelineModel como producto del entrenamiento de un modelo de aprendizaje de máquina se da lugar al pipeline que probará el modelo, Figura 31, el mismo tendrá los mismos pasos que el de entrenamiento donde se invocará al método tranform() con el *dataset* de prueba. Así tanto PipelineS y PipelineModel aseguran que los datos de entrenamiento y de prueba atraviesen los pasos de procesamiento de características en forma idéntica.

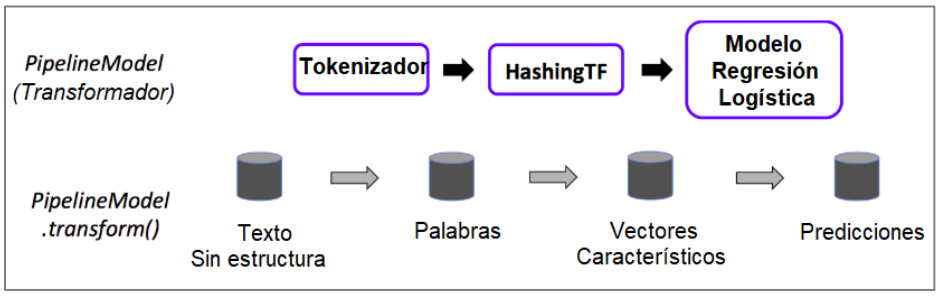


Figura 31. Pipeline de Prueba del Modelo ML

Fuente: Adaptado de (Apache Spark, 2019)

1.1.8.6.2. Extracción, transformación y selección de características

Los algoritmos que se usan con características se dividen en cuatro grupos:

- Extracción de características; extrae características a partir de datos sin estructura. Los siguientes métodos son útiles para datos en texto: TF-IDF, Word2Vec, CountVectorizer, FeatureHasher,
- Transformación de características; escala, convierte o modifica las características. Los siguientes métodos son útiles para datos en texto: Tokenizer, StopWordsRemover, n-gram.
- Selección; selecciona un subconjunto de características a partir de un conjunto mucho más grande.
- Hashing Localmente Sensible (LSH); esta clase de algoritmos combina aspectos de transformación de características con otros algoritmos.

1.1.8.6.3. Regresión logística

La regresión logística es un método popular para predecir una respuesta categórica, es un caso especial de los modelos lineales generalizados que predice la probabilidad de una salida. En spark.ml la regresión logística binomial se usa para predecir salidas binarias, está implementada por la clase LogisticRegressionModel tanto en Scala, Java y Python.

```
from pyspark.ml.classification import LogisticRegression

# Cargar datos de entrenamiento
training = spark.read.format("libsvm").load("data/mllib/sample_libsvm_data.txt")

lr = LogisticRegression(maxIter=10, regParam=0.3, elasticNetParam=0.8)

#Entrenar el modelo
lrModel = lr.fit(training)

#Imprimir los coeficientes e intercepción para regresión logística
print("Coefficients: " + str(lrModel.coefficients))
print("Intercept: " + str(lrModel.intercept))
```

Figura 32. Spark LogisticRegression en Python

Fuente: Adaptado de (Apache Spark, 2019)

1.1.8.6.4. Selección del modelo y afinamiento

MLlib tiene herramientas para afinar tanto los algoritmos de aprendizaje de máquina y la lógica de datos o *Pipeline*. La selección del modelo o *tuning* es

el uso de datos para encontrar el mejor modelo o parámetro para una tarea dada, se puede realizar tanto para un *EstimatorS* individual como *PipelineS* que incluya varios pasos de caracterización, y algoritmos. MLlib soporta la selección del modelo usando herramientas como *CrossValidator* y *TrainValidationSplit*. Ambos requieren de un Estimador o Pipeline, un conjunto de parámetros *ParamMapS* en donde buscar y un *Evaluator*, que es la métrica para medir que tan bien un modelo entrenado se desempeña con los datos de prueba. Lo que estas herramientas en líneas generales es dividir los datos de entrada en *datasets* de entrenamiento y prueba, por cada par de (entrenamiento, prueba) iteran a través del conjunto de *ParamMapS* y por cada *ParamMap* entrenan el *Estimator* usando estos parámetros, una vez obtenido el Modelo entrenado, lo evalúan usando el *Evaluator*. Luego se selecciona el Modelo que haya producido el mejor funcionamiento con el conjunto de parámetros.

El *Evaluator* puede ser un *RegressionEvaluator* para problemas de regresión, un *BinaryClassificationEvaluator* para datos binarios.

Cross-Validation, *CrossValidator* computa la métrica de evaluación promedio para el número de pares de datasets (entrenamiento, prueba) producidos por dividir el dataset original en pliegues (*folds*) producto del entrenamiento del modelo a través del *Estimador*. Después de identificar el mejor ParamMap, CrossValidator finalmente re-entrena el Estimator usando el mejor ParamMap y todo el dataset.

Train-Validation Split, TrainValidationSplit permite el afinamiento de hiper-parámetros, sólo evalúa cada combinación de parámetros una sola vez, a diferencia de CrossValidation que lo hace varias veces, por lo que es menos costoso de realizar aunque su efectividad está relacionada directamente al tamaño del dataset de entrenamiento. Además, TrainValidationSplit crea solamente un par de datasets (entrenamiento, prueba) usando el parámetro trainRatio que si es igual a 0.75 generaría un dataset que representa el 75% para el entrenamiento del modelo y 25% para la prueba.

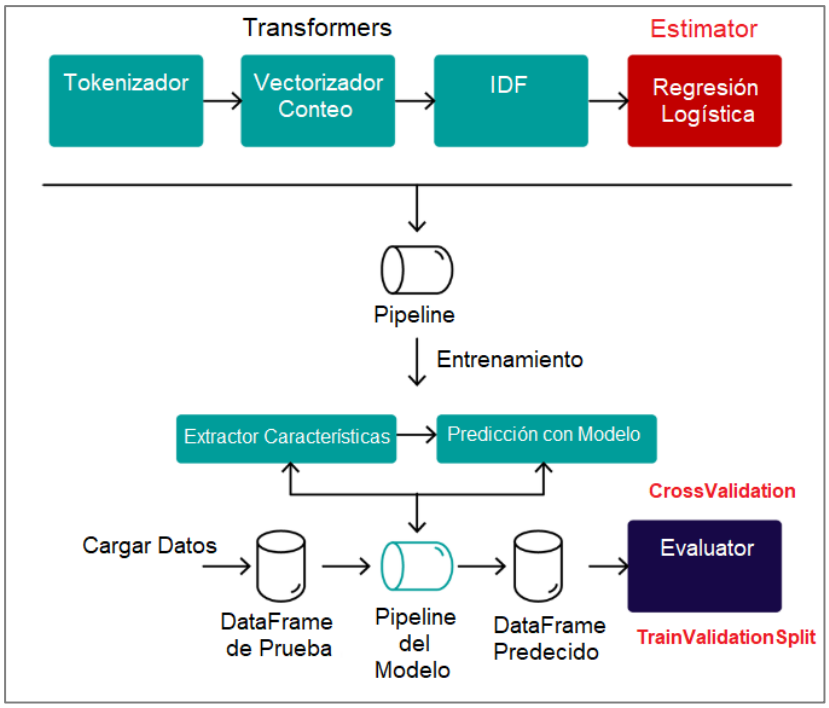


Figura 33. Predicción y Evaluación del Modelo

Fuente: Adaptado de (AI Zone, 2019)

1.2. Antecedentes

Los siguientes antecedentes muestran las investigaciones realizadas sobre análisis de opinión en Twitter usando el framework Spark:

El trabajo de Nodarakis *et al.* (2016) presenta una solución de gran escala de análisis sentimental en el framework distribuido Spark para análisis de opinión en datos de Twitter, en el que las tareas se ejecutan en forma paralela y distribuida usando trabajos MapReduce, el algoritmo de aprendizaje supervisado utilizado es de un clasificador binario y multi-clase kNN ya implementado en MapReduce que explota los hashtags, así como los emoticones que se encuentren dentro de un tuit como etiquetas sentimentales, y continua hacia un procesamiento de análisis sentimental de diversos tipos de sentimiento sin construir un lenguaje sentimental o cualquier anotación manual. Adicionalmente, la clasificación hace uso de *filtros Bloom* para compactar el tamaño de almacenamiento del conjunto de datos intermedios para incrementar el rendimiento del algoritmo de aprendizaje y disminuir el uso de nodos en el sistema distribuido de la solución. Probaron que el sistema de clasificación es eficiente, robusto y escalable en un cluster de computación de 4 nodos (1 como maestro y 3 como esclavos) que ejecutaron Spark 1.4.1 cuya configuración usada fue de 12 núcleos ejecutores y uno como driver para procesar

942,188 tuits que contenían hashtags y 1’337,508 tuits que contenían emoticones resultado de las tareas de recolección, limpieza y análisis de tuits publicados entre Noviembre 2014 a Agosto 2015. La evaluación del clasificador binario fue realizada usando el método de validación cruzada de 10 dobles para medir la precisión del clasificador. Y además se mostró que la solución escala linealmente.

El trabajo de investigación de Baltas *et al.* (2016) implementa un sistema de análisis sentimental usando el framework distribuido Spark y su API MLlib usando algoritmos de aprendiza de máquina y técnicas de procesamiento de lenguaje natural. Introducen pasos de pre-procesamiento para mejorar el analizador sentimental. Los algoritmos de clasificación son de aprendizaje supervisado de tipo binario y ternario. Además analizaron el efecto del tamaño del conjunto de datos o *dataset* y las características de entrada en el cambio de precisión del clasificador causado por el tamaño del conjunto de datos de entrenamiento.

El trabajo de investigación de Svyatkovskiy *et al.* (2016) implementa el pipeline o cola lógica de procesamiento de texto distribuido en DataFrames de Spark y una interfaz de programación en Scala para evaluar el desempeño de Apache Spark en problemas de aprendizaje de máquina con datos intensivos de texto referidos a leyes publicadas por las legislaturas de los Estados Unidos; así explican los desafíos y estrategias del procesamiento de datos sin estructura, formato de datos para almacenamiento y acceso eficiente, y procesamiento de grafos en escala. El marco de trabajo expuesto y utilizado se basa en la serialización Avro, Spark ML, GraphFrames y el conjunto Histogrammar para analizar como el lenguaje Scala se integra al eco sistema de Hadoop.

Además se ha considerado las investigaciones de análisis de opinión relevantes de Twitter que han demarcado su principio y su actualidad:

La primera investigación de análisis de opinión en Twitter se expone en el artículo de Go *et al.* (2009), en el que introducen el enfoque para clasificar el sentimiento automáticamente de tuits en el microblogging Twitter como positivo y negativo con respecto a un término de consulta. Los resultados que obtienen se basan en el uso de algoritmos de aprendizaje de máquina para la clasificación de sentimiento usando supervisión distante, donde los datos de entrenamiento son mensajes de Twitter con emoticones, que son utilizados como etiquetas relevantes del corpus. Aplicaron los algoritmos de aprendizaje de máquina: Redes Bayesianas, Entropía Máxima y SVM

alcanzando una precisión de 80% cuando fueron entrenados con mensajes que incluían emoticones. Además describen los pasos de procesamiento necesarios para alcanzar una precisión alta; muestran la forma de recolectar tuits con ayuda de la API de Twitter; caracterizan a los tuits por su tamaño, modelo de lenguaje y dominio; establecen la necesidad reducir sus características como quitar enlaces o URLs, letras repetidas, re-tuits y palabras vacías; finalmente exploraron los modelos con extractores de características como unigramas, bigramas, y parte del discurso, concluyendo que la inclusión de etiquetas del parte del discurso no agregan utilidad a la precisión de los modelos utilizados.

La tendencia actual sobre análisis de opinión o análisis de sentimiento se estudia en el artículo de Rosenthal *et al.* (2017) donde exponen las tareas de análisis de sentimiento en Twitter realizadas en el quinto concurso SemsEval del International Workshop on Semantic Evaluation, de la Asociación Lingüística Computacional; entre las tareas expuestas están la de la clásica identificación del sentimiento general de tuits, el sentimiento sobre un tema con clasificación en escala de dos a cinco puntos y la cuantificación de la distribución del sentimiento sobre un tema a través de un número de tuits a escala de dos a cinco puntos también; el estudio muestra la introducción de corpus en el idioma árabe y del clásico inglés de eventos que fueron tendencia en Twitter durante setiembre a noviembre del 2016 para el idioma árabe y de diciembre 2016 a enero 2017 para el idioma en inglés. Encontraron que los 48 equipos participantes usaron métodos basados en aprendizaje profundo y redes neuronales como CNN y LSTM, combinaciones de redes neuronales con métodos supervisados lineales como SVM, y los clásicos métodos de Entropía Máxima, Regresión Logística, Random Forest, Redes Bayesianas; además observaron que el software usado incluyó a Python (con librerías sklearn y numpy), Java, TensorFlow, Weka, NLTK, Keras, Theano, y Stanford CoreNLP; finalmente muestran que de los 48 equipos participantes, 39 han publicado un artículo de investigación sobre su desempeño en el concurso de análisis de opinión usando el corpus validado del concurso.

CAPÍTULO II

PLANTEAMIENTO DEL PROBLEMA

2.1. Identificación del problema

El uso creciente de las redes sociales, el correo electrónico, los mensajes instantáneos de texto, los *chats* en tiempo real, los tuits, han propiciado el nacimiento de una disciplina muy nueva dentro del análisis de textos: el análisis de sentimiento o sentimental (*sentiment analysis*), también conocido como análisis de opinión. En el caso de los Big Data, donde referencias muy seguras consideran que más del 80% de los datos son no estructurados, o tienen la forma de texto, se pretende integrarlos de modo que puedan ser útiles en el proceso de obtención de valor de los grandes volúmenes de datos (Aguilar, 2016).

Las personas ahora son sensores subjetivos del mundo real, lo perciben y expresan su opinión en forma de texto, y masivamente (Zhai y Massung, 2016), a través de las redes sociales interactúan instantáneamente publicando sus opiniones sobre el evento que acontece, sobre lo que les gustó o disgustó. Twitter es una de esas redes sociales donde se acumulan las opiniones, desde que fue lanzado en el año 2006, se convirtió en un fenómeno masivo, en el 2013 en la plataforma que tenía más de 200 millones de usuarios activos en 33 idiomas diferentes, quienes publicaban más de 400 millones de tuits por día (Weller *et al.*, 2014).

Twitter es un gran depósito de datos masivos, sobre todo de textos cortos de 140 caracteres por su característica particular de microblogging; el que a partir del año 2008 ha sido objeto de numerosos estudios (Pang y Lee, 2008) , especialmente en lo referente al subjetivismo, sentimiento u opinión dentro de los tuits, los algoritmos de máquina supervisados y no supervisados para comprender la opinión del texto han sido utilizados con diversas variantes, Sentiment140 es un claro ejemplo de su popularidad en las

investigaciones, así como el emblemático estudio de (Go *et al.*, 2009) que estableció los pasos de este proceso. Sin embargo todas estos estudios se han ido realizado siguiendo una arquitectura centralizada, tanto las herramientas de minería de datos como Weka, ScikitLearn y otros muy populares funcionan de forma centralizada o en un solo espacio de memoria reduciendo su alcance a *datasets* pequeños y negando el uso de aquellos que sobrepasaran el espacio de memoria o que son los de Big Data.

En tiempos de Ciencia de los datos, Big Data y el Aprendizaje de Máquina es necesario migrar los modelos de análisis de opinión centralizadas a distribuidos para poder tratar con la masividad de los datos que se producen en Twitter (Baltas *et al.*, 2016) garantizando a su vez que el *performance* alcanzado sea lo suficientemente bueno como ya lo son los de arquitectura centralizada durante todos estos años y de esa manera se pueda hallar valor en los datos que se producen en las redes sociales cuando los usuarios producen grandes volúmenes de publicaciones en respuesta a eventos trascendentales de nivel mundial como es el caso de los mundiales de futbol.

2.2. Enunciado del problema

Por lo tanto se realizó la siguiente formulación del problema: ¿Es posible que el analizador de opinión del microblogging Twitter por la clasificación al mundial de futbol Rusia-2018 de la selección peruana de futbol usando el framework Spark alcance un performance razonablemente bueno como los de arquitectura centralizada?

2.3. Justificación

Aplicar algoritmos de análisis de opinión o clasificación del sentimiento como parte de los procesos de minería de datos, y específicamente del aprendizaje de máquina en forma distribuida para aprovechar los grandes conjuntos de datos semi estructurados existentes en el microblogging Twitter, que se producen en acontecimientos o eventos que promueven publicaciones masivas de tuits como por ejemplo la clasificación al mundial Rusia-2018 de la selección peruana de futbol y realizar analíticas, permite demostrar que las tareas de aprendizaje de máquina, como las de clasificación/predicción, pueden realizarse a nivel distribuido manteniendo un performance razonablemente bueno como las que se realizan en forma centralizada. Así, esta investigación contribuye en las áreas de Ciencia de los Datos al aplicar algoritmos de aprendizaje de máquina para grandes volúmenes de datos en una arquitectura distribuida, y formaliza una metodología de

implementación en tareas de análisis de opinión o clasificación de sentimiento en Big Data, los mismos que pueden ser y son utilizados por las organizaciones interesadas en implementar sistemas de inteligencia de negocios para la toma de decisiones, marketing y sistemas de recomendación en la utilización efectiva de los datos masivos y externos a la empresa, organización o institución que existen en las redes sociales.

Como utilidad metodológica, esta investigación permite conocer el proceso y las técnicas utilizadas en el proceso de análisis de opinión en sistemas distribuidos, como son el pre-procesamiento de datos, modelamiento, y verificación del modelo en el *framework* Spark; así como el proceso de construcción del *dataset* específico para las tareas de análisis o clasificación binaria a partir de datos históricos de Twitter dado un evento mundialmente trascendente como un mundial de futbol.

2.4. Objetivos

2.4.1. Objetivo general

Analizar la opinión de los tuits publicados por la clasificación al mundial de futbol Rusia-2018 de la selección peruana de futbol en el microblogging Twitter usando el framework Spark.

2.4.2. Objetivos específicos

- Construir el *dataset* de tuits para el análisis de opinión en el microblogging Twitter por la clasificación al mundial de futbol Rusia-2018 de la selección peruana de futbol.
- Pre-procesar el dataset, entrenar y evaluar el modelo de análisis de opinión en el framework Spark para clasificar las opiniones en el microblogging Twitter por la clasificación al mundial Rusia-2018 de la selección peruana de Futbol.

2.5. Hipótesis

2.5.1. Hipótesis general

El analizador de opinión clasifica adecuadamente los tuits del microblogging Twitter por la clasificación al mundial de futbol Rusia 2018 de la selección peruana de futbol usando el framawork Spark.

2.5.2. Hipótesis específicas

- 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” de análisis de opinión de Twitter.
- La exactitud del modelo de análisis de opinión del microblogging Twitter por la clasificación al mundial de futbol Rusia 2018 de la selección peruana de futbol en el framework Spark es mayor a la exactitud promedio de los modelos de SemEval-2017 Task 4: Message Polarity Classification.

CAPÍTULO III

MATERIALES Y MÉTODOS

3.1. Lugar de estudio

La investigación se realizó en la Universidad Nacional del Altiplano de Puno, donde se construyó el dataset para el análisis de opinión, se desarrolló el analizador de opinión del microblogging Twitter por la clasificación al mundial Rusia-2018 de la selección peruana de futbol usando el framework Spark en un ambiento local.

3.2. Población

La población para esta investigación estuvo compuesta por todos los tuits históricos publicados en el microblogging Twitter que tuvieran relación con el evento de clasificación al mundial de Rusia-2018 de la selección peruana futbol en español desde Octubre del 2015 hasta Noviembre del 2017, siendo por lo tanto N = Infinito, ya que se desconoce cuántos tuits en total han sido publicados en este periodo de tiempo, y se supone que es un muy grande por la envergadura de tuits publicados en la red social Twitter.

3.3. Muestra

La selección de la muestra fue de tipo no probabilístico y se utilizó el muestreo casual o incidental en el que el investigador selecciona directamente los individuos de la población que sean accesibles. Así, la muestra se compuso por 500 tuits históricos por 20 intervalos de fechas combinados con 162 *hashtags y 80 cuentas de usuario* de Twitter que tuviesen relación con el evento de clasificación al mundial de Rusia-2018 de la selección peruana de futbol en español. Las fechas elegidas siguieron el calendario clasificatorio de la FIFA Ronda 1 y Play-Off Copa Rusia 2018 (Tabla 2, y Tabla 3)con un margen de dos días

anteriores y dos días siguientes al día del partido de futbol de la selección peruana (5 fechas por cada partido), los *hashtags* y las cuentas de usuario elegidos se corresponden al evento de clasificación al mundial Rusia-2018 mostrado en la Tabla 4.

Tabla 2

Fechas de Ronda clasificatoria al Mundial Rusia 2018 - Perú

Nro	Match	Fecha	Fecha Búsqueda
1	Colombia - Perú	09 Oct 2015	Del 07 Oct 2015 al 11 Oct 2015
2	Perú – Chile	13 Oct 2015	Del 11 Oct 2015 al 15 Oct 2015
3	Perú - Paraguay	13 Nov 2015	Del 11 Nov 2015 al 15 Nov 2015
4	Brasil – Perú	17 Nov 2015	Del 15 Nov 2015 al 19 Nov 2015
5	Perú - Venezuela	14 Mar 2016	Del 12 Mar 2016 al 16 Mar 2016
6	Uruguay – Perú	29 Mar 2016	Del 27 Mar 2016 al 31 Mar 2016
7	Bolivia – Perú	01 Set 2016	Del 30 Ago 2016 al 03 Set 2016
8	Perú – Ecuador	06 Set 2016	Del 04 Set 2016 al 08 Set 2016
9	Perú - Argentina	06 Oct 2016	Del 04 Oct 2016 al 08 Oct 2016
10	Chile – Perú	11 Oct 2016	Del 09 Oct 2016 al 13 Oct 2016
11	Paraguay - Perú	10 Nov 2016	Del 08 Nov 2016 al 12 Nov 2016
12	Perú – Brasil	15 Nov 2016	Del 13 Nov 2016 al 17 Nov 2016
13	Venezuela - Perú	23 Mar 2017	Del 21 Mar 2017 al 25 Mar 2017
14	Perú – Uruguay	28 Mar 2017	Del 26 Mar 2017 al 30 Mar 2017
15	Perú – Bolivia	31 Ago 2017	Del 29 Ago 2017 al 02 Set 2017
16	Ecuador – Perú	05 Set 2017	Del 03 Set 2017 al 07 Set 2017
17	Argentina - Perú	05 Oct 2017	Del 03 Oct 2017 al 07 Oct 2017
18	Perú - Colombia	10 Oct 2017	Del 08 Oct 2017 al 12 Oct 2017

Fuente: Adaptado de (FIFA.com, 2018)

Tabla 3

Fechas de Play-Off clasificatoria al Mundial de Rusia 2018 - Perú

Nro	Match	Fecha Match	Fecha Búsqueda
1	Nueva Zelanda Perú	11 Nov 2017	Del 09 Nov 2017 al 13 Nov 2017
2	Perú – Nueva Zelanda	15 Nov 2017	Del 13 Nov 2017 al 17 Nov 2017

Fuente: Adaptado de (FIFA.com, 2018)

Tabla 4

Hashtags Trending Topic y cuentas de usuario con relación a la selección peruana de futbol en Twitter.

Nro	Hashtag	Cuenta de usuario
1	#VamosPerú	@TuFPF
2	#LaHinchadaDelPerú	@E_FLEISCHMAN
3	#LaBlanquiroja	@blanquirroja
4	#ArribaPerú	@marca
5	#VamosPeruanos	@CONMEBOL
6	#Rusia2018	@Odriozola9
7	#LaMejorHinchadaDelMundo	@DIRECTVSports
8	#15NOV2017	@ComadoSvr1986
9	#Selección	@18andrecarrillo
10	#LocalesEnTodasPartes	@Universitario

Fuente: Adaptado de (Trendogate.com, 2018)

3.4. Método de la investigación

La presente investigación es del tipo experimental tecnológico porque manipula directamente la variable independiente que son los tuits históricos relacionados al evento de clasificación al mundial de Rusia-2018 de la selección peruana de futbol en español para medir sus efectos en la variable dependiente que es el modelo de análisis de opinión del microblogging Twitter, este método se aplicó con el propósito de establecer las conclusiones y generalizar los resultados de la investigación en forma cuantitativa.

3.5. Descripción detallada de métodos por objetivos específicos

Para el logro del objetivo específico 01: “Construir el dataset de tuits para el análisis de opinión en el microblogging Twitter por la clasificación al mundial de futbol Rusia-2018 de la selección peruana de futbol”, se empleó parte de la metodología propuesta por (McCreadie *et al.*, 2012) para construir legalmente un cuerpo de datos de Twitter, desarrollado en colaboración con la Conferencia de Recuperación de Texto 2011 microblogging track (TREC 2011) y Twitter. Esta metodología consiste de dos pasos, primero recuperar los tuits a través de un arrastrador HTTP asíncrono que descarga cada tuit individualmente desde el sitio de Twitter.com y reconstruye los tuits en el formato JSON o CSV sin usar directamente la API de Twitter como se muestra en la Figura 34, y

segundo etiquetar manualmente cada tuit por los participantes del TREC 2011, para finalmente crear un dataset que sigue el esquema estándar de datos propuesto por (Go *et al.*, 2009) en el dataset "Sentiment140".

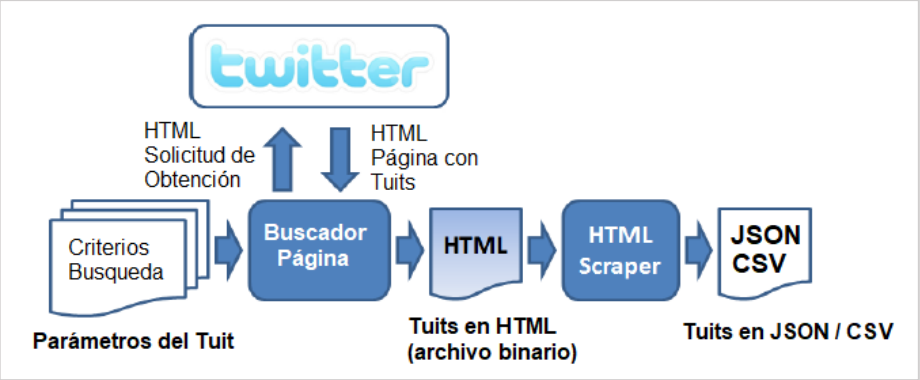


Figura 34. Recuperación de tuits a través de un Scraper HTML

Fuente: Adaptado de (McCreadie *et al.*, 2012)

Las herramientas utilizadas para alcanzar este objetivo fueron el uso de la API open source GetOldTweets-python (Jefferson, 2016) que arrastra tuits históricos pasados y el lenguaje de programación Python 3.4; para construir el sistema de etiquetado manual se usó un ambiente de programación Web basado en: PHP, JQuery, Bootstrap 4; junto a la base de datos MySQL 5.7.

Para cumplir con el objetivo 02: "Pre-procesar el *dataset*, entrenar y evaluar el modelo de análisis de opinión aplicando el framework Spark para clasificar las opiniones en el microblogging Twitter por la clasificación al mundial Rusia-2018 de la selección peruana de Futbol", se empleó la metodología del proceso de clasificación de opinión o sentimiento. La Figura 35 ilustra este proceso.

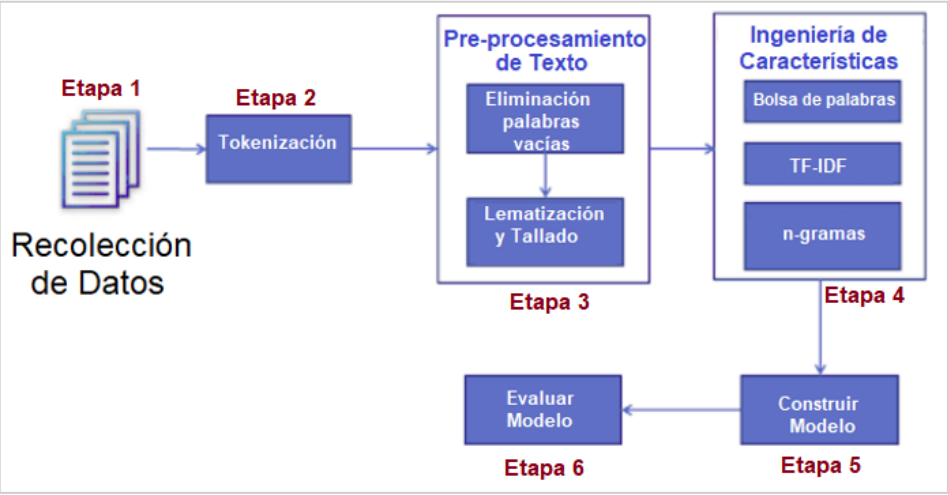


Figura 35. Etapas del proceso de análisis de opinión

Fuente: Adaptado de (Liu, 2012)

Las herramientas utilizadas fueron las que proporciona el framework Spark basadas en: SparkSession, DataFrame y los algoritmos de la API MLlib de PySpark: TransformerS, EstimatorS, PipelineStageS, Tokenizer, HashingTF, Regex, StopWordsRemover, LogisticRegression, LogisticRegressionModel, CrossValidator, RegressionEvaluator, BinaryClassificationEvaluator, TrainValidatorSplit, IDF, Word2Vec, CountVectorizerModel, y NGram. Cada una de estas herramientas se utiliza en el proceso de pySpark que muestra la Figura 36.

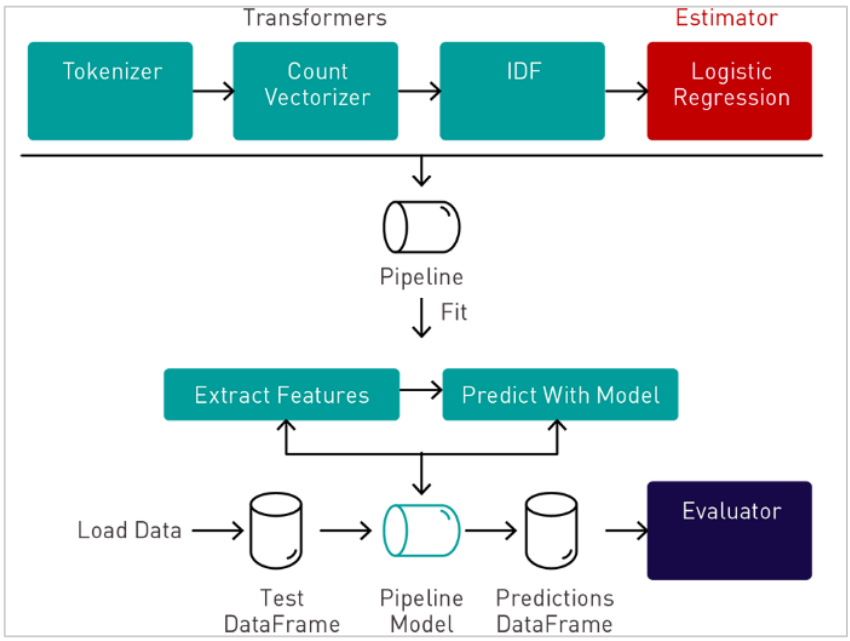


Figura 36. Proceso de análisis de sentimiento con Spark MLlib

Fuente: Adaptado de (AI Zone, 2019)

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

En este capítulo se presenta los resultados y discusión de cada uno de los objetivos específicos de la investigación.

4.1. Resultados conforme al objetivo específico 1

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

4.1.1. Recuperar tuits

Se utilizó el Lenguaje de Programación Python 3.4 y la API GetOldTweets-python de (Jefferson, 2016) para desarrollar el script que arrastró los tuits pasados o históricos que cumplieron con los siguientes parámetros de búsqueda: intervalo de fecha de búsqueda y hashtag o intervalo de fecha de búsqueda y nombre de cuenta. Los parámetros de búsqueda se fijaron del siguiente modo:

- **Intervalo de Fecha de búsqueda**, se obtuvo 11 intervalos de fechas de búsqueda correspondientes al calendario clasificatorio FIFA Ronda 1 y Play-Off Copa Rusia 2018 de la selección peruana (FIFA.com, 2018), la Tabla 5 la resume.
- **Hashtag**, se obtuvo **162** hashtags relacionados a la selección peruana de futbol a través de la exploración manual de tuits de la cuenta oficial de la selección peruana de futbol en Twitter.com @FPF y la aplicación Web Trendogate.com (Trendogate.com, 2018), la Tabla 7 muestra un resumen de los 35 *hashtags* más relevantes según la fecha correspondiente al encuentro de futbol y los hashtags restantes se encuentran en el Anexo 01.

Tabla 5

Intervalos de Fechas de búsqueda del calendario FIFA Ronda 1 y Play-Off Copa Rusia 2018 selección peruana de futbol.

Nro	Intervalo de Fecha Búsqueda	Fecha Match	Días
1	Del 07 Oct 2015 al 15 Oct 2015	09 Oct 2015 &13 Oct 2015	09
2	Del 11 Nov 2015 al 19 Nov 2015	13 Nov 2015&17 Nov 2015	09
3	Del 12 Mar 2016 al 16 Mar 2016	14 Mar 2016	05
4	Del 27 Mar 2016 al 31 Mar 2016	29 Mar 2016	05
5	Del 30 Ago 2016 al 08 Set 2016	01 Set 2016 & 06 Set 2016	10
6	Del 04 Oct 2016 al 13 Oct 2016	06 Oct 2016 &11 Oct 2016	10
7	Del 08 Nov 2016 al 17 Nov 2016	10 Nov 2016 &15 Nov 2016	10
8	Del 21 Mar 2017 al 30 Mar 2017	23 Mar 2017 &28 Mar 2017	10
9	Del 29 Ago 2017 al 07 Set 2017	31 Ago 2017 &05 Set 2017	10
10	Del 03 Oct 2017 al 12 Oct 2017	05 Oct 2017 &10 Oct 2017	10
11	Del 09 Nov 2017 al 17 Nov 2017	11 Nov 2017 &15 Nov 2017	09

Fuente: Elaborado según (FIFA.com, 2018)

- **Nombre de cuenta**, se obtuvo 80 cuentas de usuario de Twitter relacionados a la selección peruana de futbol recolectados de de cuentas oficiales de la Federación Peruana de Futbol, clubes deportivos, periodistas deportivos, programas deportivos, periódicos deportivos y jugadores de la selección peruana; la Tabla 6 resume las 10 cuentas más relevantes según el número de seguidores; las cuentas de usuarios restantes se encuentran en el Anexo 01.

Tabla 6

Cuentas de usuario con mayor cantidad de seguidores de la selección peruana de futbol

Nro	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
6	DIRECTV Sports	@DIRECTVSports	612,7 K	Programa deportivo
7	André Carrillo	@18andrecarrillo	371,8 K	Jugador selección
8	Comando SVR	@ComandoSvr1986	138,1 K	Barra Alianza Lima
9	La Blanquirroja	@blanquirroja	33,5 K	Barra tribuna sur

Tabla 7

Hashtags Trending Topic según fecha calendario FIFA Ronda 1 y Play-Off Copa Rusia 2018 selección peruana de futbol.

Nro	Match	Fecha	HashTags Trending Topic
1	Colombia – Perú	09 Oct 2015	#PerdimosComoSiempre
2	Perú – Chile	13 Oct 2015	#PorLaBlanquiroja, #PeruVsChile
3	Perú – Paraguay	13 Nov 2015	#HoyGanaPeruPor, #AlientoCocaCola
4	Brasil – Perú	17 Nov 2015	#LeTengoFeA, #SiPeruGanaYo
5	Perú – Venezuela	14 Mar 2016	#VamosPeru, #PeruVsVenezuela
6	Uruguay – Perú	29 Mar 2016	#ClaroQueSiSePuede, #UruguayvsPeru
7	Bolivia – Perú	01 Set 2016	#PeruVsBolivia, #ArribaPeru #SeleccionPeruana
8	Perú – Ecuador	06 Set 2016	#SumarAntesQueRestar, #YaFue
9	Perú – Argentina	06 Oct 2016	#PeruvsArgentina, #ArribaPeru
10	Chile – Perú	11 Oct 2016	#ParaQuePeruGane, #ChileNoNosGanaPor
11	Paraguay – Perú	10 Nov 2016	#ArribaPeru, #Rusia2018
12	Perú – Brasil	15 Nov 2016	#Moscú, #NadieNosPara
13	Venezuela – Perú	23 Mar 2017	#CHONGOPERU4ANO
14	Perú – Uruguay	28 Mar 2017	#AúnTengoEsperanzas
15	Perú – Bolivia	31 Ago 2017	#SiLaBlanquirojaVaAlMundial #SiPeruHace6Puntos
16	Ecuador – Perú	05 Set 2017	#SiPerúGanaPrometo, #ContigoPerú
17	Argentina – Perú	05 Oct 2017	#selecciónperuana #estaesmicábalaparahoy
18	Perú – Colombia	10 Oct 2017	#Islandia, #EstánPasandoCosas #PorQueYoCreoEnTi

Fuente: Adaptado de (FIFA.com, 2018) y (Trendogate.com, 2018)

Script de Arrastre de Raw Data

El Script Python elaborado para el arrastre de tuits se compone de dos métodos: descargar y orquestar, el primer método se encarga de arrastrar N tuits y genera un archivo CSV, **descargar(ofn, ds, du, qs, mt)** donde los parámetros determinan: **ofn** el nombre del archivo de salida, **ds** la fecha inicial del intervalo de búsqueda, **du** la fecha final del intervalo de búsqueda, **qs** la palabra de consulta y **mt** el número máximo de tuits arrastrados. El segundo método u orquestador se encarga de repetir el arrastre para todas las combinaciones

buscadas entre fechas y hashtags o fechas y cuentas, **main()** recupera la lista de intervalos de fechas y hashtags a partir de archivos csv y ejecuta repetitivamente el método descargar combinando ambos criterios de búsqueda. Las figuras 37 y 38 muestran ambas partes del Script resultante del programa **descargarTuits.py** desarrollado para estas tareas.

```
11  def descargar(ofn, ds, du, qs, mt):
12      try:
13          tweetCriteria = got.manager.TweetCriteria()
14          outputFileName = ofn
15          tweetCriteria.since = ds
16          tweetCriteria.until = du
17          tweetCriteria.querySearch = qs
18          tweetCriteria.maxTweets = mt
19          outputFile = codecs.open(outputFileName, "w+", "utf-8")
20          outputFile.write('username;date;retweets;favorites;text;geo;mentions;hashtags;id;
            permalink')
21          print('Searching...\n')
22
23          def receiveBuffer(tweets):
24              for t in tweets:
25                  outputFile.write(('\n%s;%s;%d;%d;"%s";%s;%s;%s;"%s";%s' % (t.username,
                    t.date.strftime("%Y-%m-%d %H:%M"), t.retweets, t.favorites, t.text, t.geo,
                    t.mentions, t.hashtags, t.id, t.permalink)))
26              outputFile.flush()
27              print('More %d saved on file...\n' % len(tweets))
28          got.manager.TweetManager.getTweets(tweetCriteria, receiveBuffer)
29      except arg:
30          print('Arguments parser error, try -h' + arg)
31      finally:
32          outputFile.close()
33          print('Done. Output file generated "%s".' % outputFileName)
```

Figura 37. Método que descarga tuits con consulta de búsqueda del programa descargarTuits.py

```
35  def main():
36      with open('fechaBusqueda.csv') as f:
37          reader = csv.reader(f)
38          listaFechas = list(reader)
39      with open('hashtags.csv') as f:
40          reader = csv.reader(f)
41          listaHashtags = list(reader)
42      for fecha in listaFechas:
43          f1 = fecha[0]
44          f2 = fecha[1]
45          t = 1
46          for hashtag in listaHashtags:
47              h = hashtag[0]
48              archivo = "tuits-" + f1 + "-" +str(t)+ ".csv"
49              descargar(archivo,f1,f2,h,500)
50              t = t + 1
```

Figura 38. Método orquestador para descargar múltiples tuits del programa descargarTuits.py

La ejecución del programa **descargarTuits.py** para intervalos de fecha de búsqueda y hashtags, así como intervalos de fecha de búsqueda y cuentas de usuario dio los siguientes resultados:

Tabla 8

Número de archivos CSV obtenidos por el arrastre de tuits históricos según criterios de búsqueda

Criterio de Búsqueda	Número de archivos de salida
Intervalo de Fecha y hashtags	1784
Intervalo de Fecha y cuenta	880
Total	**2664**

Tabla 9

Número total de tuits arrastrados y recuperados exitosamente

Criterio	Categoría de Éxito	Archivos	Tuits Arrastrados
Intervalo de Fecha de búsqueda y Hashtags	500 Tuits	249	124500
	Menos de 500 Tuits	369	92250
	0 Tuits	1166	0
	Sub Total	**1784**	**216750**
Intervalo de Fecha de búsqueda y Cuentas	500 Tuits	67	33500
	Menos de 500 Tuits	504	126000
	0 Tuits	309	0
	Sub Total	**880**	**159500**
	Total	**2664**	**376250**

- Se obtuvieron 2664 archivos en formato CSV, 1784 archivos producto del arrastre de tuits por la combinación de intervalo de fecha y hashtags, 880 archivos producto del arrastre de tuits por la combinación de intervalo de fecha y cuenta de usuario, resumidos en la Tabla 8. Cada archivo CSV contiene una instancia que representa a tuit con 10 datos descritos en la Figura 39.

```
instanciaTuitDescargado= {
    username:   "nombre de usuario",
    date:       "fecha y hora de publicación",
    retweets:   "numero de re-publicaciones",
    favorites:  "numero de likes a la publicacion",
    text:       "texto del tuit",
    geo:        "locacion del tuit",
    mentions:   "URLs agregadas al tuit",
    hashtags:   "etiquetas contenidas en el tuit",
    id:         "Identificador del usuario",
    permalink:  "URL del tuit"
}
```

Figura 39. Esquema de los datos que componen un tuit arrastrado y recuperado

- Cada arrastre exitoso acumuló 500 tuits como máximo por cada combinación de parámetros de búsqueda, mientras que los arrastres medianamente exitosos acumularon menos de 500 tuits, y los fallidos no acumularon ningún tuit. Así se llegó a arrastrar aproximadamente **376250 tuits**, 216750 de la combinación intervalo de fecha de búsqueda y hashtags; y 159500 de la combinación intervalo de fecha de búsqueda y cuenta de usuario. La Tabla 9 lo resume.
- Por lo tanto se ha conseguido 376250 tuits como *Raw Data* o datos en estado sin procesamiento.

4.1.2. Etiquetado

Para etiquetar los tuits y completar la construcción del dataset se desarrolló una aplicación web que permite etiquetar manualmente a un usuario cada tuit en las categorías: positivo y negativo. Se utilizó herramientas del ambiente de programación web compuesto por PHP, la API Bootstrap 4.0, y la API JQuery. La aplicación web recupera un archivo CSV generado por el arrastramiento de tuits o Raw Data, enlista los tuits hallados en el archivo CSV y permite que el usuario etiquete manualmente el tuit en positivo, negativo y neutro. Sólo los tuits etiquetados como positivos y negativos se guardaron en una tabla llamada Tuits de la base de datos **bdTuits**, para luego generar un archivo CSV global que contuvo todos los tuits etiquetados. La Figura 40 muestra la arquitectura de la aplicación web que se desarrolló.

La base de datos estuvo gestionada por MySQL 5.7, donde se creó la base de datos **bdTuits** y la tabla **tuits** con el mismo esquema del dataset “Sentiment 140” construido por (Go *et al.*, 2009) que acumuló los tuits etiquetados manualmente y a partir de esta tabla se generó el archivo CSV que contiene al dataset resultante

necesario para el entrenamiento del modelo de clasificación en Spark. La figura 41 describe el esquema de la tabla **tuits**.

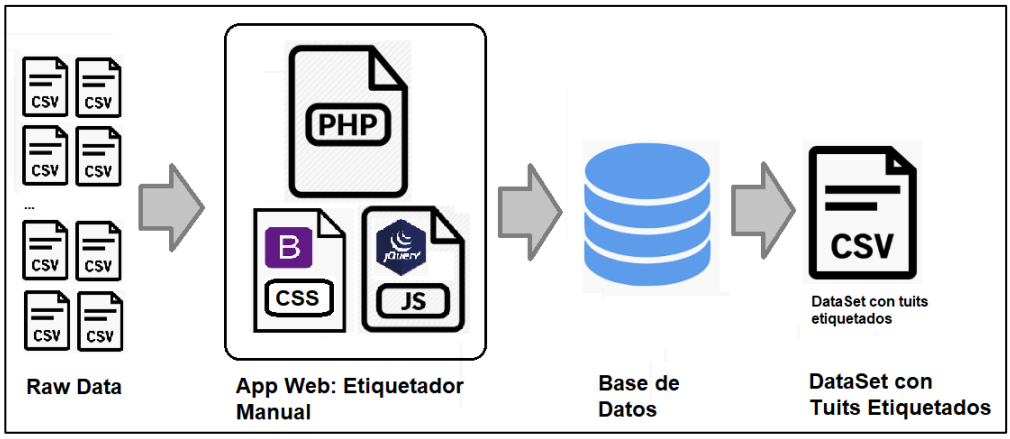


Figura 40. Arquitectura de la Aplicación web para el etiquetado manual

```
use bdtuit;
describe tuits;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
polaridad	int(11)	YES		NULL	
id_tuit	bigint(20)	NO	PRI	NULL	
fecha	varchar(18)	YES		NULL	
consulta	varchar(100)	YES		NULL	
usuario	varchar(100)	YES		NULL	
texto	text	YES		NULL	

Figura 41. Esquema de la tabla tuits según el esquema de Sentiment140

Fuente: Elaborado a partir de (Go *et al.*, 2009)

La Figura 42 muestra la interfaz de la aplicación web desarrollado para etiquetar tuits manualmente por terceros. Está compuesto por un menú a partir del que se enlistan los tuits contenidos en un archivo csv, el enlistado se produjo uno por uno del que se muestra la fecha y hora de publicación del tuit, el hashtag, el nombre de la cuenta del usuario, y el texto que contenía el tuit, por defecto cada tuit ya está etiquetado como neutro, solo cuando el usuario lo haya marcado como positivo o negativo será guardado en la tabla **tuits** de la base de datos con la polaridad etiquetada manualmente. En el Anexo 02 se detalla el funcionamiento de la aplicación de etiquetado manual.

Así, se etiquetó 5000 tuits manualmente, del que se obtuvo 1500 tuits con polaridad negativa y 3500 con polaridad positiva, a partir del cual se construyó un dataset final en formato CSV simétrico de 3000 tuits, 1500 tuits negativos y 1500 tuits

positivos llamado **PeruARuisa2018.csv**. La Tabla 10 describe el esquema del dataset resultante.

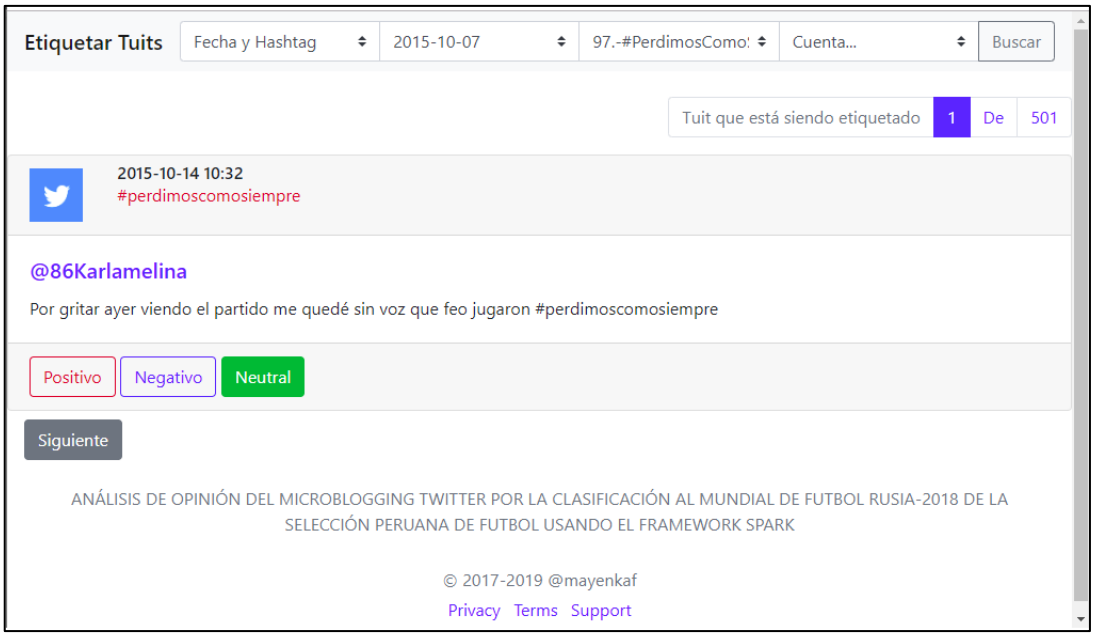


Figura 42. Interfaz de usuario de la Aplicación web para el etiquetado manual de tuits

Tabla 10

Estructura del Dataset PeruARusia2018

Atributo	Descripción	Valores
Polaridad	Etiqueta del tuit	0: Negativo 4: Positivo
Id_tuit	Identificador del tuit	Valores numéricos
Fecha	Fecha de publicación del tuit	Valores cadena
Consulta	Hashtag utilizado dentro del tuit	Valores cadena
Usuario	Nombre del usuario	Valores cadena
Texto	Texto contenido por el tuit etiquetado	Valores cadena

4.1.3. Discusión

La construcción del dataset PeruARusia2018.csv fue realizado siguiendo la metodología de arrastre y obtención de tuits históricos de (McCreadie *et al.*, 2012), que proponen arrastrar directamente los tuits de un determinado evento y tiempo usando un extractor HTTP asíncrono sin usar directamente un servicio de obtención de datos desde Twitter; esta propuesta permitió superar las restricciones de acceso

a tuits históricos cuando se usa directamente la API Rest de Twitter (Twitter.com, 2018) y las limitaciones en el acceso al número de tuits por petición, así la obtención de los tuits relacionados al evento de clasificación de la selección peruana al mundial de Rusia se realizó con la ejecución del programa **descargarTuits.py** que hizo uso de la API GetOldTweets de (Jefferson, 2016).

Otro aspecto importante es que el dataset PeruARusia2018.csv debe garantizar el estándar de los datos que están contenidos dentro de un dataset como entrada para un modelo de clasificación binario en Spark, por lo que se ha seguido el esquema del dataset o corpus Sentiment140 desarrollado por (Go *et al.*, 2009) quienes lo construyeron y propusieron como el primer dataset de análisis sentimental y que durante este tiempo ha sido el más utilizado por la comunidad investigadora como entrada para los algoritmos de análisis de opinión o clasificación de sentimiento, donde se han probado diversos métodos de clasificación y ha permitido la comparación de los performances hallados. Así las características de los datos que contiene el dataset PeruARusia2018.csv son los mismos del esquema de datos del dataset Sentiment140.

Finalmente el dataset PeruRusia2018.csv debe estar ya etiquetado para ser utilizado en modelos de aprendizaje supervisado de aprendizaje de máquina, por lo que se siguió la metodología de etiquetado desarrollado por (McMinn *et al.*, 2013) que proponen etiquetar el dataset en base a la opinión de terceros o de personas para así garantizar la relevancia y calidad del dataset aunque sea el más laborioso de realizar porque es totalmente manual; así el etiquetado manual realizado para desarrollar el dataset PeruARusia2018.csv se ha realizado a través de una aplicación web que ha permitido a usuarios humanos etiquetalos, y que a su vez, ha permitido descartar tuits spam, retuits, y tuits neutros o sin subjetivismo. Es así que el dataset PeruARusia2018.csv ha reunido las características que la hace adecuada para ser usado dentro de un modelo de clasificación de sentimiento de aprendizaje de máquina.

4.2. Resultados conforme al objetivo específico 2

Pre-procesar el *dataset*, entrenar y evaluar el modelo de análisis de opinión aplicando el framework Spark para clasificar las opiniones en el microblogging Twitter por la clasificación al mundial Rusia-2018 de la selección peruana de Futbol.

4.2.1. Pre-procesar el dataset

Se utilizó el framework Spark 2.3.1, 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. Se realizó los siguientes pasos para cumplir con esta tarea:

a. Se importó las librerías SparkSession, DataFrame, Regex_Replace, LogisticRegression, Tokenizer, HashingTF, StopWordsRemover del lenguaje pySpark del framework Spark 2.3.1. La Figura 43 la resume.

```
#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()
```

Figura 43. Librerías utilizadas en pySpark y creación de la sesión Spark para la aplicación "Analisis_Opinion_PeruARusia"

b. Se cargó y leyó el *dataset* "PeruARusia2018.csv" en el DataFrame **tuits_peruArusia**, se infirió el esquema a partir de la primera línea que contenía los nombres de las columnas del dataset. La Figura 44 lo muestra.

```
#cargar datos en un DataFrame
tuits_peruArusia = spark.read.csv("datos/PeruARusia2018.csv", inferSchema = True, header = True)
tuits_peruArusia.show(5)
```

```
+---------+--------------+---------------+--------------------+------------+--------------------+
|polaridad|        id_tuit|          fecha|            consulta|     usuario|               texto|
+---------+--------------+---------------+--------------------+------------+--------------------+
|        0|120151009097021|2015-10-09 20:00|#PerdimosComoSiempre|    @efatatv|la llorona http:/...|
|        0|120151009097025|2015-10-09 16:00|#PerdimosComoSiempre|    @efatatv|Nunca podrán http...|
|        4|120151009097026|2015-10-09 15:53|#PerdimosComoSiempre|   @mapiabsi|Jugamos como nunc...|
|        4|120151009097033|2015-10-09 10:59|#PerdimosComoSiempre|@pierlicious|#PerdimosComoSiem...|
|        4|120151009097034|2015-10-09 10:20|#PerdimosComoSiempre|  @Wenlizita|#PerdimosComoSiem...|
+---------+--------------+---------------+--------------------+------------+--------------------+
only showing top 5 rows
```

Figura 44. Carga del dataset PeruARusia2018.csv en un DataFrame

c. Se creó el DataFrame **tuitsData** con las columnas útiles como entrada del modelo de aprendizaje, estas incluyeron la polaridad etiquetada que se

convirtió de tipo String a Int y el texto del tuit del DataFrame tuits_peruArusia. La Figura 45 lo muestra.

```
#seleccionar los datos necesarios para el modelo de aprendizaje
tuitsData = tuits_peruArusia.select(col("polaridad").cast("Int").alias("label"),"texto")
tuitsData.show(2)
```

```
+-----+--------------------+
|label|               texto|
+-----+--------------------+
|    0|la llorona http:/...|
|    0|Nunca podrán http...|
+-----+--------------------+
only showing top 2 rows
```

Figura 45. Selección de las columnas polaridad y texto como datos útiles

d. Se realizó las siguientes tareas de pre-procesamiento en el DataFrame **tuitsData**, sobre la columna "texto" y se obtuvo un texto limpio y que pudo ser procesado numéricamente por el modelo de clasificación binario. Esto incluyó:

- Remplazo de vocales atildadas por vocales sin tilde, se utilizó la función **translate** que remplazó caracteres por otros. La Figura 45 la resume.

```
#1.-limpiando texto, remplazando vocales atildadas por vocales simples
dataSinTilde = tuitsData.select(translate(col("texto"),"áéíóú","aeiou")\
                                .alias("textoSinTilde")\
                                ,col("label"))
```

Figura 46. Limpieza del texto de vocales con tilde

- Eliminación de palabras que representan usuarios y etiqueta hashtag, se utilizó la función regexp_replace para remplazar el patrón de una expresión regular que representa a una palabra de un usuario o una etiqueta hashtag en Twitter, los mismos que comienzan con @ o # son seguidos de cualquier otro carácter alfanumérico por un carácter vacío "". La Figura 47 lo muestra.

```
#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"))
```

Figura 47. Limpieza del texto de palabras que representan usuarios o etiquetas hashtag de Twitter

- Eliminación de URLs, se utilizó la función regex_replace para remplazar el patrón de una expresión regular que representa a una URL, los mismos que

son combinaciones de palabras alfanuméricas, y el símbolo "/" por un carácter vacío "". La Figura 48 lo muestra.

```
#4.-limpiando texto, 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"))

#5.-limpiando texto, quitando tatus/... word.word.com/ http:// https://
patron="http://|https://|[A-Za-z0-9]+/[A-Za-z0-9]+\
|[A-Za-z0-9]+.[A-Za-z0-9]+.[A-Za-z0-9]+/[A-Za-z0-9]+"
dataSinURLs = dataSinURL.select(regexp_replace(col("textoSinURL"),patron,"")\
                                .alias("textoSinURLs")\
                                ,col("label"))
```

Figura 48. Limpieza del texto de palabras que representan URLs

- Eliminación de signos de puntuación y espacios, se utilizó la función regex_replace para remplazar el patrón de una expresión regular que representa a símbolos y signos de puntuación como la doble comilla, punto, punto y coma, dos puntos, signos de interrogación, admiración, igualdad, y espacios por un carácter vacío "". La Figura 49 lo muestra.

```
#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 49. Limpieza del texto de caracteres que representan signos de puntuación, interrogación, admiración y espacios en blanco

```
dataLimpio.show(10, False)
```

```
+----------------------------------------------------------------------------------------------------+-----+
|text                                                                                                |label|
+----------------------------------------------------------------------------------------------------+-----+
|la llorona ow                                                                                       |0    |
|Nunca podran ow                                                                                     |0    |
|Jugamos como nunca y Pero igual hasta las finales con ustedeslos veremos en Recien empezamos        |1    |
| asi es el deporte pero aun seguimos alentando                                                      |1    |
| eso es historia repetida pero nunca perderemos la FE                                               |1    |
|Jugamos con rebeldia garra inteligencia pero                                                        |1    |
|Yo ante cada partido de Peru frente a cualquier otro equipo y la familia me dicen mala onda         |0    |
| pero al menos metimos miedo                                                                        |0    |
| Ahora a esperar el clasico del pacifico Al menos perderemos la esperanza Grau perdio un barquito|0    |
| Lo de siempre creemos pero la realidad nos mata los sueños El futbol duele                         |0    |
+----------------------------------------------------------------------------------------------------+-----+
only showing top 10 rows
```

Figura 50. Tuits cuyo texto están limpios

La Figura 50 muestra los 10 primeros tuits con el texto limpiado en la columna **text**, esta es una vista del dataset transformado por el proceso de pre-procesamiento de datos en la columna texto para que pueda servir como entrada al paso de vectorización o conversión numérica del texto.

4.2.2. Entrenar el modelo

Se construyó el modelo de aprendizaje supervisado de clasificación binaria basado en Regresión Logística usando la librería de Machine Learning de Spark MLlib, se siguió los siguientes pasos para entrenar y probar el modelo:

a. Se dividió el dataset **dataLimpio** que representa a todo el corpus con pre-procesamiento del dataset original **PeruARusia2018.csv** en dos partes: 70% del total de datos para que formen parte de los datos de entrenamiento **tuitsEntrenamiento** y 30% del total de datos para que formen parte de los datos de prueba **tuitsPrueba**, cada ítem de cada grupo ha sido elegido usando la función **randomSplit.** La Figura 51 lo muestra.

```
#Dividir el dataset en entrenamiento=70% y prueba=30%
tuitsDividido = dataLimpio.randomSplit([0.7, 0.3])
tuitsEntrenamiento = tuitsDividido[0]
tuitsPrueba = tuitsDividido[1]
nroTuitsEntrenamiento = tuitsEntrenamiento.count()
nroTuitsPrueba = tuitsPrueba.count()
print("Tuits de Entrenamiento :",nroTuitsEntrenamiento, "Tuits de Prueba :",nroTuitsPrueba)
```

```
Tuits de Entrenamiento : 1991 Tuits de Prueba : 813
```

Figura 51. División del dataset en datos de entrenamiento y prueba

b. Se tokenizó en palabras el texto de la columna **text**, esta separación se realizó por espacios en blanco. La Figura 52 lo muestra.

```
#Preparar Datos: Tokenizacion
tokenizer = Tokenizer(inputCol = "textoSinEsp", outputCol="textoPalabras")
tokenizedData = tokenizer.transform(dataSinEspacios)
tokenizedData.select("textoPalabras").show(5, False)
```

```
+--------------------------------------------------------------------------------------------------------------------+
|textoPalabras                                                                                                       |
+--------------------------------------------------------------------------------------------------------------------+
|[la, llorona, ow]                                                                                                   |
|[nunca, podran, ow]                                                                                                 |
|[jugamos, como, nunca, y, pero, igual, hasta, las, finales, con, ustedeslos, veremos, en, recien, empezamos]        |
|[, asi, es, el, deporte, pero, aun, seguimos, alentando]                                                            |
|[, eso, es, historia, repetida, pero, nunca, perderemos, la, fe]                                                    |
+--------------------------------------------------------------------------------------------------------------------+
only showing top 5 rows
```

Figura 52. Tokenización del texto en palabras individuales

c. Se ha quitado las palabras sin utilidad o StopWords para eliminar aquellas palabras que no agregan valor a partir del idioma español que tiene por defecto la librería StopWordsRemover de Spark, estas palabras son artículos, adverbios, tabs. La Figura 53 lo muestra.

```
#Preparar Datos: quitar palabras sin importancia de spanish
spanishStopWords = StopWordsRemover.loadDefaultStopWords("spanish")
swr = StopWordsRemover(stopWords = spanishStopWords, inputCol = tokenizer.getOutputCol(), outputCol="palabras")
swrTuitsData = swr.transform(tokenizedData)
swrTuitsData.select("palabras").show(6, False)
```

```
+------------------------------------------------------------------------+
|palabras                                                                |
+------------------------------------------------------------------------+
|[llorona, ow]                                                           |
|[nunca, podran, ow]                                                     |
|[jugamos, nunca, igual, finales, ustedeslos, veremos, recien, empezamos]|
|[, asi, deporte, aun, seguimos, alentando]                              |
|[, historia, repetida, nunca, perderemos, fe]                           |
|[jugamos, rebeldia, garra, inteligencia]                                |
+------------------------------------------------------------------------+
only showing top 6 rows
```

Figura 53. Eliminación de los stopwords del idioma Español

d. Se transformó las palabras del dataset de entrenamiento en representación numérica usando un transformador y la función **HashingTF,** el mismo que halla la Frecuencia de las palabras presente en el texto del tuit. La Figura 54 lo muestra.

```
#Vectorizar las palabras en características numéricas
from pyspark.ml.feature import HashingTF
hashTF = HashingTF(inputCol="palabras",outputCol="features")
numericTrainData = hashTF.transform(tuitsEntrenamiento).select(col("Etiqueta").alias("label"),"palabras","features")
numericTrainData.show(n=3)
```

```
+-----+--------------------+--------------------+
|label|            palabras|            features|
+-----+--------------------+--------------------+
|    0|[, &lt&lt&lt, q, ...|(262144,[23574,48...|
|    0|[, *gareca, agrad...|(262144,[40991,71...|
|    4|[, 1982, joven, a...|(262144,[4407,214...|
+-----+--------------------+--------------------+
only showing top 3 rows
```

Figura 54. Conversión de palabras a números usando HashingTF

e. Se construyó un modelo de clasificación binaria basado en regresión logística con las siguientes características: la entrada de entrenamiento es la columna

"features", y la de salida es "label", el modelo entrenó en 10 iteraciones con un parámetro de regularización de 0.01. La Figura 55 lo muestra.

```
#Entrenar el modelo con los datos de entrenamiento
lr = LogisticRegression(labelCol="label",featuresCol="features",maxIter=10, regParam=0.01)
modelo = lr.fit(numericTrainData)
print("El modelo esta entrenado!")
```

Figura 55. Modelo de Clasificación basado en Regresión Logística

4.2.3. Evaluación del modelo

Se ha evaluado el modelo a través del cálculo de la exactitud, precisión y recuperación del modelo de clasificación binaria basado en Regresión Logística usando el *dataset* de Prueba para medir el *performance* del analizador. Así se ha realizado los siguientes pasos:

a. Se transformó la columna "palabras" a su representación numérica tal como se hizo con el *dataset* de entrenamiento usando la función HashingTF. La Figura 56 lo muestra.

```
#Vectorizar los datos de prueba
numericTest = hashTF.transform(tuitsPrueba).select(col("Etiqueta").alias("label"),"palabras","features")
numericTest.show(n = 3)
```

```
+-----+--------------------+--------------------+
|label|            palabras|            features|
+-----+--------------------+--------------------+
|    4|[, 2017, fecha, t...|(262144,[14316,38...|
|    0|[, cualquier, lad...|(262144,[19576,78...|
|    0|[, gusta, futbol,...|(262144,[34243,71...|
+-----+--------------------+--------------------+
only showing top 3 rows
```

Figura 56. Transformación numérica de los datos de Prueba

b. Se calculó las medidas de evaluación del modelo de clasificación binario: exactitud (*accuracy*), recuperación (*recall*), y precisión (*precision*), estas medidas muestran el *performance* del modelo. La figura 57 muestra que la exactitud alcanzada es del 83.51%, la precisión del 82.30% y la recuperación o sensibilidad del 89.28%.

```
true_positives = predictionFinal.filter('label==1 and prediction==1.0').count()
print(true_positives)

true_negatives = predictionFinal.filter('label==0 and prediction==0.0').count()
print(true_negatives)

false_positives = predictionFinal.filter('label==0 and prediction==1.0').count()
print(false_positives)
false_negatives = predictionFinal.filter('label==1 and prediction==0.0').count()
print(false_negatives)

recall = float(true_positives)/(true_positives + false_negatives)
print("Recall :", recall)
precision = float(true_positives)/(true_positives + false_positives)
print("Precision :", precision)
accuracy = float(true_positives + true_negatives)/(totalData)
print("Accuracy :", accuracy)

Recall : 0.8928571428571429
Precision : 0.823045267489712
Accuracy : 0.8351783517835178
```

Figura 57. Calculo de medidas de evaluación del modelo de clasificación basada en Regresión Logística

4.2.4. Discusión

El pre-procesamiento de los datos de la columna que representa al texto de cada tuit dentro del dataset ha sido realizado dentro del propio framework Spark siguiendo los pasos de estas tareas que menciona (Go *et al.*, 2009) que es estandarizar las palabras, lo que incluye quitar tildes, signos de puntuación, admiración, URLs, nombre de cuentas, y como del mismo modo (Liu, 2012) proponen eliminar aquellas palabras que no son palabras del propio idioma que está siendo utilizado dentro del tuit ya que no representan adecuadamente el sentimiento del tuit, como es la eliminación de artículos, preposiciones, conectores que a diferencia de (Nodarakis *et al.*, 2016) que no contemplan eliminar estas palabras especiales sino que la normalizan por palabras como URL, REF y TAG y se mantienen presentes para el modelo de clasificación. Por otra parte las tareas de tokenización y eliminación de palabras sin valor o stopwords se han realizado conforme a Go, Liu y Nodarakis cuando describen los pasos de procesamiento de datos con menor ruido, en cuanto a la tokenización se han separado las palabras por espacios en blanco y la eliminación de stopwords se han realizado usando el estándar de la librería StopWordsRemover para el idioma español que ofrece Spark, no se ha adicionado otras características del idioma.

En cuanto al modelo de clasificación binaria basada en Regresión Logística, se ha aplicado el recomendado por (Zhai y Massung, 2016) para realizar modelos de clasificación de sentimiento o de opinión en texto. Este modelo ideal que describe

Zhai es un referente adecuado para hallar una marca inicial en cuanto a la clasificación binaria; así la aplicación del modelo LogisticRegression que ofrece Spark ha sido entrenado en 10 repeticiones con un parámetro de regularización 0.01 de la función objetivo sobre el 70% del dataset como parte de los datos de entrenamiento vectorizado con HashingTF o cálculo del TF-IDF.

Las medidas de evaluación del modelo de análisis de opinión o sentimental es uno de tipo binario, estas incluyeron una exactitud alcanzada del 83.51% lo que significa que el modelo tiene una predicción correcta de 8 tuits de cada 10, sean estos de clase positivo o negativo, también alcanzó una precisión del 82.30% lo que significa 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, esto sugiere que el modelo predice correctamente casi 9 de cada 10 tuits correctamente de todos los casos predichos positivamente. Si comparamos estos resultados, se puede observar que son mayores a los obtenidos en la Sub Tarea A: "Message Polarity Classification" que (Rosenthal *et al.*, 2017) exponen como medidas de evaluación halladas en el SemEval-2017 Task 4, si bien es cierto que estos modelos no han entrenado con el mismo dataset, se puede decir también como que las marcas de evaluación alcanzadas están dentro de las marcas que cualquier modelo de clasificación binaria debe alcanzar, es decir estar sobre el 70% en exactitud, esto nos indica que el modelo se comporta adecuadamente como otros modelos que han usado librerías de arquitectura centralizada.

4.3. Prueba de hipótesis

Para la prueba de hipótesis: El analizador de opinión clasifica adecuadamente los tuits del microblogging Twitter por la clasificación al mundial de futbol Rusia 2018 de la selección peruana de futbol usando el framawork Spark, se ha utilizado como método la prueba de hipótesis sobre la media, y se han planteado las siguientes hipótesis estadísticas:

Con respecto a la construcción del dataset para el análisis de opinión del microblogging Twitter por la clasificación al mundial de futbol Rusia 2018 de la selección peruana de futbol.

Tabla 11

Comparación de Características del dataset Sentiment140 y PeruARusia2018

Dataset	12 características de un dataset	Medida Cuantitativa
Sentiment 140	Archivo externo CSV	Si = 1
	Formato de Texto UTF-8	Si = 1
	Balanceado en 2 clases (Tuits positivos es igual a Tuits negativos)	Si = 1
	Balanceado en 3 clases (Hay tantos tuis positivos como negativos y nulos)	No = 0
	Esquema contiene: Id, Fecha Tuit, Consulta, Usuario que publicó el tuit, Texto intacto del Tuit, Etiqueta del Tuit	Si = 6
	Nro de Tuits = 1600000	Si = 1
	El texto del tuit contiene emoticonos	No = 0
	Total	**10 de 12**
PeruARusia2018	Archivo externo CSV	Si = 1
	Formato de Texto UTF-8	Si = 1
	Balanceado (Tuits positivos es igual a Tuits negativos)	Si = 1
	Balanceado en 3 clases (Hay tantos tuis positivos como negativos y nulos)	No = 0
	Esquema contiene: Id, Fecha Tuit, Consulta, Usuario que publicó el tuit, Texto intacto del Tuit, Etiqueta del Tuit	Si = 6
	Nro de Tuits = 1600000	No = 0
	El texto del tuit contiene emoticonos	No = 0
	Total	**9 de 12**

Hipótesis Nula:

H_0: 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 no tiene las mismas características estándar del dataset "Sentiment140" de análisis de opinión de Twitter. $\boldsymbol{H_0}: \overline{X}_{\boldsymbol{dsPeruARusia}} \neq \mu_{dsSentiment140}$

Hipótesis Alternativa:

H_1: 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" de análisis de opinión de Twitter. $\boldsymbol{H_1}: \boldsymbol{\bar{X}_{dsPeruARusia}} = \mu_{dsSentiment140}$

Nivel de Significancia:

Se eligió el nivel de significancia de 0,02 o 2% de error.

$$\alpha = 0{,}02 = 2\% \text{ y } GL = 11$$

Se utilizó la distribución t

$t_\alpha = t_{0,02} = -2.718\ y\ 2.718.$

Zona de rechazo y regla de decisión:

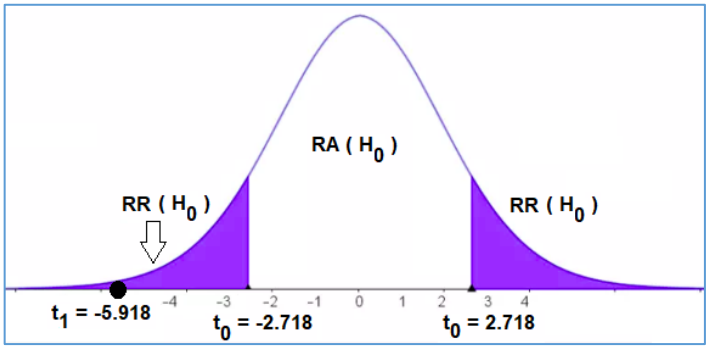
Se usó el estadístico de prueba para $n = 12$; n < 30; Donde $n = 12$ es el número de características de un dataset:

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

Estadística de Prueba:

$t_1 = \frac{9-10}{0.577/\sqrt{12}} = -5.477$; Se ha usado los datos proporcionados por la tabla 11 en el remplazo de la ecuación.

Y el diagrama con las zonas de rechazo laterales derecha e izquierda, el nivel de significancia y t_1:



Luego de realizada la prueba estadística, rechazamos la Hipótesis nula por lo tanto se acepta la Hipótesis alterna, 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" de análisis de opinión de Twitter, por lo que es adecuado para utilizarla en modelos de clasificación de texto.

Con respecto a la hipótesis específica 02: La exactitud del modelo de análisis de opinión del microblogging Twitter por la clasificación al mundial de futbol Rusia 2018 de la selección peruana de futbol en el framework Spark es mayor a la exactitud promedio de los modelos de SemEval-2017 Task 4: Message Polarity Classification. Se ha planteado las siguientes hipótesis estadísticas:

Hipótesis Nula:

H0: la exactitud del modelo de análisis de opinión del microblogging Twitter por la clasificación al mundial de futbol Rusia-2018 de la selección peruana de futbol usando el framework Spark es menor o igual a la exactitud de los modelos de SemEval-2017 Task 4: Message Polarity Classificaction; $\boldsymbol{H_0}: \overline{\boldsymbol{X}}_{\boldsymbol{aot}} \leq \boldsymbol{\mu}_{\boldsymbol{semEval}}$

Hipótesis Alternativa:

H1: la exactitud del modelo de análisis de opinión del microblogging Twitter por la clasificación al mundial de futbol Rusia-2018 de la selección peruana de futbol usando el framework Spark es mayor a la exactitud de los modelos de SemEval-2017 Task 4: Message Polarity Classification. $\mathbf{H_1}: \overline{\mathbf{X}}_{\mathbf{aot}} > \boldsymbol{\mu}_{\mathbf{semEval}}$

Nivel de Significancia:

Se eligió el nivel de significancia de 0,05 o 5% de error.

$$\alpha = 0{,}05 = 5\%$$

Se utilizó la distribución Z

$$Z_\alpha = Z_{0{,}05} = 1.64$$

Zona de rechazo y regla de decisión:

Se usó el estadístico de prueba para $n = 813$; n > 30:

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

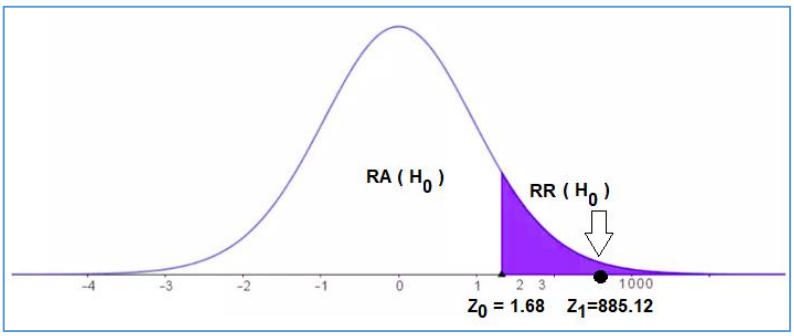
Estadística de Prueba:

$Z_1 = \frac{679-529}{4.83/\sqrt{813}} = 885.12$; Se ha usado los datos proporcionados por la tabla 12 en el remplazo de la ecuación.

Tabla 12. *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.

Y el diagrama con la zona de rechazo unilateral derecha y el nivel de significancia sería igual a:



Luego de realizada la prueba estadística, rechazamos la Hipótesis nula por lo tanto se acepta la Hipótesis alterna, así que la exactitud del modelo de análisis de opinión del

microblogging Twitter por la clasificación al mundial de futbol Rusia-2018 de la selección peruana de futbol usando el framework Spark es mayor a la exactitud de los modelos de SemEval-2017 Task 4: Message Polarity Classification, por lo que su performance es adecuado.

CONCLUSIONES

- El analizador de opinión del microblogging Twitter por la clasificación al mundial de futbol Rusia-2018 de la selección peruana de futbol usando el framework 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 del 89.28%.

- El dataset de tuits construido "PeruARusia2018.csv" para realizar análisis de opinión del microblogging Twitter por la clasificación al mundial Rusia-2018 de la selección peruana de futbol es adecuado para entrenar modelos de aprendizaje de tipo clasificación binaria, el cual cumple con las condiciones de los datasets estándares de la comunidad científica en análisis de opinión o sentimiento.

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

RECOMENDACIONES

- Se recomienda probar otros modelos de aprendizaje supervisado en el área de clasificación de opinión o sentimiento con el dataset PeruARusia2018.csv que esta investigación ha construido para mejorar el rendimiento de analizadores de opinión en el idioma español que se desempeñen en sistemas distribuidos como Spark.

- Se recomienda utilizar técnicas de análisis sintáctico y marcador gramatical como un paso adicional para comprobar si el análisis de opinión o sentimiento mejora la precisión de los mismos en el idioma español, ya que esta aproximación no se ha utilizado en esta investigación.

- Se recomienda integrar el análisis de opinión a sistemas de recomendación e inteligencia de negocio para la mejor toma de decisiones.

BIBLIOGRAFÍA

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 de https://dzone.com/articles/streaming-machine-learning-pipeline-for-sentiment

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

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

Chambers, B., & 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 de https://www.fifa.com/worldcup/preliminaries/southamerica/all-matches.html

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.

Jurafsky, D., & Manning, C. (2012). Natural language processing. *Instructor, 212*(998), 3482.

Leskovec, J., Rajaraman, A., & 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.

Lopez Briega, R. E. (2016). Machine Learning con Python - Sobreajuste.

McCreadie, R., Soboroff, I., Lin, J., Macdonald, C., Ounis, I., & 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., & 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.

Mughal, M. J. H. (2018). Data Mining: Web Data Mining Techniques, Tools and Algorithms: An Overview. *Information Retrieval, 9*(6).

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., & 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., & 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.

Twitter.com. (2018). Documentación de la API Rest de Twitter.

Weller, K., Bruns, A., Burgess, J., Mahrt, M., & Puschmann, C. (2014). *Twitter and society* (Vol. 89): Peter Lang.

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

Zafarani, R., Abbasi, M. A., & 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.

ANEXOS

Anexo 1. Construcción del Dataset

a. Fechas calendario Ronda 01 y Play-Off Copa Rusia 2018, FIFA.com

Nro	Match	Fecha Match	Rango de Fecha Búsqueda
1	Colombia – Perú	09 Oct 2015	Del 07 Oct 2015 al 11 Oct 2015
2	Perú – Chile	13 Oct 2015	Del 11 Oct 2015 al 15 Oct 2015
3	Perú – Paraguay	13 Nov 2015	Del 11 Nov 2015 al 15 Nov 2015
4	Brasil – Perú	17 Nov 2015	Del 15 Nov 2015 al 19 Nov 2015
5	Perú - Venezuela	14 Mar 2016	Del 12 Mar 2016 al 16 Mar 2016
6	Uruguay – Perú	29 Mar 2016	Del 27 Mar 2016 al 31 Mar 2016
7	Bolivia – Perú	01 Set 2016	Del 30 Ago 2016 al 03 Set 2016
8	Perú – Ecuador	06 Set 2016	Del 04 Set 2016 al 08 Set 2016
9	Perú – Argentina	06 Oct 2016	Del 04 Oct 2016 al 08 Oct 2016
10	Chile – Perú	11 Oct 2016	Del 09 Oct 2016 al 13 Oct 2016
11	Paraguay – Perú	10 Nov 2016	Del 08 Nov 2016 al 12 Nov 2016
12	Perú – Brasil	15 Nov 2016	Del 13 Nov 2016 al 17 Nov 2016
13	Venezuela - Perú	23 Mar 2017	Del 21 Mar 2017 al 25 Mar 2017
14	Perú – Uruguay	28 Mar 2017	Del 26 Mar 2017 al 30 Mar 2017
15	Perú – Bolivia	31 Ago 2017	Del 29 Ago 2017 al 02 Set 2017
16	Ecuador - Perú	05 Set 2017	Del 03 Set 2017 al 07 Set 2017
17	Argentina - Perú	05 Oct 2017	Del 03 Oct 2017 al 07 Oct 2017
18	Perú - Colombia	10 Oct 2017	Del 08 Oct 2017 al 12 Oct 2017

Ronda 1.

Nro	Match	Fecha Match	Rango de Fecha Búsqueda
1	Nueva Zelanda Perú	11 Nov 2017	Del 09 Nov 2017 al 13 Nov 2017
2	Perú – Nueva Zelanda	15 Nov 2017	Del 13 Nov 2017 al 17 Nov 2017

Play-Off

b. HashTags sobre la selección peruana de futbol

Nro	Hashtag	Nro	Hashtag
1	#10añosalentando	39	#EstamosDeVuelta
2	#15NOV2017	40	#estánpasandocosas
3	#AbrazodeGol	41	#EstánPasandoCosas
4	#alientoporque	42	#EsteAmorNoEsParaCobardes
5	#ArribaPeru	43	#EstoNoAcaba
6	#arribaperu	44	#FloroTeLlevaConLaSelección

7	#ArribaPerú	45	#FoxSportsPeru
8	#ArribaPerúCarajo	46	#FuerzaCapitán
9	#AsiNoJuegaPerú	47	#FuerzaPaolo
10	#AúnTengoEsperanzas	48	#FutbolTotalDIRECTVPeru
11	#BanderazoBlaquirojo	49	#GainPERU
12	#Blanquiroja	50	#GanaPeruYYo
13	#CachimboMundialista	51	#GolesSiGolpesNo
14	#CanciónMundialista	52	#GraciasATodos
15	#ChileNoNosGanaPor	53	#GraciasMuchachos
16	#chongoperu4no	54	#GraciasPorTodo
17	#CHONGOPERU4NO	55	#Guerrero
18	#ClaroQueSePuede	56	#HemosVuelto
19	#ClaroQueSiSePuede	57	#HinchadaDelPerú
20	#claroquetesigoperu	58	#HinchaQueSeRespeta
21	#ConciertosBlanquirojos	59	#HoyGanaPeruPor
22	#ConFe	60	#Islandia
23	#ConLaTricolorPuesta	61	#JuntémonosParaAlentar
24	#CONTIGOPERU	62	#juntemonosporlaseleccion
25	#ContigoPeru	63	#LaBanca
26	#ContigoPerú	64	#LaBanquiroja
27	#contigoperú	65	#LaBarraMásPower
28	#ConUnTriunfoDePeru	66	#lablanquirrojadelfuturo
29	#CONVOCADOS	67	#LaBlanquirrojaEnQuito
30	#EliminatoriasCapital	68	#LaCamisetaSeLlevaEnLaGarganta
31	#eliminatoriasrusia2018	69	#LaHinchadaDelPerú
32	#EliminatoriasxCapital	70	#LaHinchadaDeTodosLosPeruanos
33	#ElNacionalEsDeLaSelección	71	#LaHinchadaMásFielDelMundo
34	#ElRegresoDelGuerrero	72	#LaHinchadaMasFielDeSudámerica
35	#EscribamosNuestraHistoria	73	#LaMejorHinchadaDelMundo
36	#ESPNPerú	74	#LaSelecciónPorATV
37	#estaesmicábalaparahoy	75	#LaVozDeTodas
38	#EstamosContigoPaolo	76	#LeTengoFeA

Hashtags

Nro	Hashtag	Nro	Hashtag
77	#LocalesEnTodasPartes	115	#PorqueYoCreoEnTi
78	#LosHinchas	116	#PreguntasParaMessi
79	#MiComboPalPartido	117	#PreparadosParaTodo
80	#MiScoreEs	118	#QuitoEsBlanquiroja

81	#ModoRusia	119	#RevoluciónBlanquiroja
82	#ModoSele	120	#RicardoGareca
83	#modosele	121	#RumboARusia2018
84	#Moscú	122	#Rusia
85	#Mundialistas	123	#RUSIA2018
86	#MundialRusia	124	#Rusia2018
87	#NadieNosPara	125	#rusia2018
88	#NosVeránVolver	126	#RusiaAlláVamos
89	#NZLvsPER	127	#RusiaSeráBlanquiroja
90	#PaoloEstaDeVuelta	128	#RutaBlanquiroja
91	#PaoloGuerrero	129	#Selección
92	#ParaEstePartidoYo	130	#SeleccionPeruana
93	#ParaQuePerúGane	131	#selecciónperuana
94	#ParenLasOrejas	132	#SiempreContigoPerú
95	#PensarComoElTigre	133	#SiLaBlanquirojaVaAlMundial
96	#PER	134	#SiNoSufrimosNoVale
97	#PerdimosComoSiempre	135	#SiPerúGanaPrometo
98	#peredototal	136	#SiPeruGanaYo
99	#PeruAlMundial	137	#siperúganayo
100	#PeruAlMundialYKeikoAlPenal	138	#SiPeruHace6Puntos
101	#PerúEnElMundial	139	#sisepuede
102	#PerúEnRusia	140	#SíSePuedePerú
103	#PerúFutbolSummit2018	141	#SomosMundialistas
104	#PERURUMBOARUSIA	142	#SomosPerúYEstamosDeVuelta
105	#PeruvsBolivia	143	#SumarAntesQueRestar
106	#PeruvsBrasil	144	#TeAmoPeru
107	#Peruvsecuador	145	#TeAmoPerú
108	#peruvsnewzealand	146	#teamoperú
109	#PerúvsUruguay	147	#TeApuestoQueEnElPartido
110	#PeruvsVenezuela	148	#TresPalabrasParaLaSelección
111	#PoemaPeru	149	#UnidosPorRusia2018
112	#PonteLaCamiseta	150	#UnSoloAliento
113	#pontelacamiseta	151	#UruguayvsPeru
114	#porqueyocreoenti	152	#vamosconfe

Hashtags

Nro	Hashtag	Nro	Hashtag
153	#VamosConTodo	158	#vamosperucarajo

154	#VamosMiSeleccion	159	#VamosPerúCarajo
155	#VamosPerú	160	#VamosPerúSiempre
156	#vamosperú	161	#VengoPorqueTeQuiero
157	#VamosPeruanos	162	#Volveremos

Hashtags

c. Cuentas de usuarios en Twitter relacionadas a la selección peruana de futbol.

Nro	Nombre	Cuenta	Seguidores	Categoría
1	Jose Chavarri	@Jose_Chavarri	36 K	Analista deportivo
2	Comando SVR	@ComandoSvr1986	138,1 K	Barra alianza lima
3	El Blog Íntimo	@ClubALoficial	11,2 K	Barra alianza lima
4	La Franja Perú	@LaFranjaPeru	4,715	Barra la franja
5	Sentimiento Blanquirrojo	@SENTIBLANQUIRRO	6,325	Barra tribuna norte
6	La Blanquirroja	@blanquirroja	33,5 K	Barra tribuna sur
7	Asociación Todo por la U	@Asoc_TodoporlaU	16 K	Barra universitario
8	Banda de Odriozola	@Odriozola9	23,9 K	Barra universitario
9	Garra 1924	@GarraucremaPeru	10,5 K	Barra universitario
10	Hinchada Crema	@hinchadacrema	16,9 K	Barra universitario
11	Club Alianza Lima	@ClubALoficial	231 K	Club deportivo
12	Club Cienciano	@Club_Cienciano	28 K	Club deportivo
13	Club Universitario de Deportes	@Universitario	863,6 K	Club deportivo
14	Deportivo Municipal	@CCDMunicipal	23,7 K	Club deportivo
15	FBC Melgar	@MelgarOficial	25,5 K	Club deportivo
16	Real Garcilazo	@SomosRealGarci	31,3 K	Club deportivo
17	San Martin	@Club_USMP	33,7 K	Club deportivo
18	Sport Boys	@sportboys	29,8 K	Club deportivo
19	Sporting Cristal	@ClubSCristal	129,9 K	Club deportivo
20	UCV Club Deportivo	@clubucv	28,3 K	Club deportivo
21	CONMEBOL.COM	@CONMEBOL	1,2 M	Confederación
22	Federación Peruana de Futbol	@TuFPF	1,1 M	Federación
23	Aldo Corzo	@Alditocorzo	108,9 K	Jugador selección
24	Andre Carrillo	@18andrecarrillo	371,8 K	Jugador selección
25	Cristian Cueva	@Cuevachris10	32,4 K	Jugador selección

Cuentas de Usuario Twitter

Nro	Nombre	Cuenta	Seguidores	Categoría
26	Edison Flores	@edisonflores135	313,8 K	Jugador selección
27	Jefferson Farfan	@JeffersonF_10	122,4 K	Jugador selección

28	Luis Advincula	@luisadvincula17	273,6 K	Jugador selección
29	Miguel Trauco	@mtrauco17	98,2 K	Jugador selección
30	Paolo Hurtado	@phurtado0712	97,9 K	Jugador selección
31	Pedro Gallese	@pedrogallese	160,2 K	Jugador selección
32	Raúl RuiDíaz	@RaulRuidiazM	307 K	Jugador selección
33	Renato Tapia Cortijo	@renatotapiac	139,9 K	Jugador selección
34	Yordy Reyna	@yordy_10	135,3 K	Jugador selección
35	Alberto Beingolea	@BeingoleaA	26,6 K	Periodista deportivo
36	Aldo Miyashiro	@ElMiyashiro	2,2 M	Periodista deportivo
37	Carlos Alberto Navarro	@tutigrilloperu	40,5 K	Periodista deportivo
38	Coki Gonzales	@cokigonzales	180,9 K	Periodista deportivo
39	Daniel Peredo M	@danielperdo17	648,3 K	Periodista deportivo
40	Eddie Fleishman	@E_FLEISCHMAN	745,5 K	Periodista deportivo
41	Gilda Arrúa	@FIFAWorldCupPER	15,6 K	Periodista deportivo
42	Gonzalo Nuñez Andrade	@gonzaperucarajo	79 K	Periodista deportivo
43	Horacion Zimmermann	@Horacon	21 K	Periodista deportivo
44	Jesus Arias	@eltankearias	430,8 K	Periodista deportivo
45	Michael Succar	@MSUCCAR	49,3 K	Periodista deportivo
46	Omar Ruiz de Somocurcio	@o_somocurcio	12,9 K	Periodista deportivo
47	Peter Arévalo	@Arevalosport12	22,6 K	Periodista deportivo
48	Ricardo Montoya	@RMontoyaDes	11, 8 K	Periodista deportivo
49	Richard De La Piedra	@crdelapiedra	26,6 K	Periodista deportivo
50	Robert Malca	@ROBERTMALCATV	10 K	Periodista deportivo
51	Sammy Sadovnik	@sadovnik1965	41 K	Periodista deportivo
52	Wily Melgarejo Ramos	@WilyMelgarejo	21,2 K	Periodista deportivo
53	Deporte Total	@dt_elcomercio	9 745	Portal deportivo
54	Deportes La Republica	@DeportesLR	29,8 K	Portal deportivo
55	Diario As	@diarioas	2,6 M	Portal deportivo
56	Diario Libero Peru	@liberoperu	2,074	Portal deportivo
57	Diario Sport	@sport	1,6 M	Portal deportivo
58	Diario trome	@tromepe	555, 1 K	Portal deportivo
59	Futbol Peruano	@futbolperuanoTR	2,6 K	Portal deportivo
60	Marca	@marca	5,2 M	Portal deportivo
61	Mundo Deportivo	@mundodeportivo	2,6 M	Portal deportivo
62	Ovación	@ovacionweb	49,5 K	Portal deportivo
63	DIRECTV Sports	@DIRECTVSports	612,7 K	Programa deportivo
64	ESPN Perú	@ESPNperu	2,740	Programa deportivo

Cuentas de Usuario Twitter

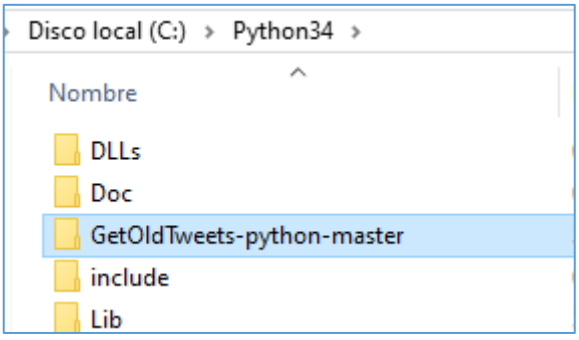
Nro	Nombre	Cuenta	Seguidores	Categoría
65	Exitosa Deportes	@Exitosadeportes	58 K	Programa deportivo
66	Fox Sports Radio Perú	@FS_RadioPeru	36,9 K	Programa deportivo
67	Futbol Como Cancha	@FutComoCancha	15,4 K	Programa deportivo
68	Futbol en América	@FAamericatv	24,3 K	Programa deportivo
69	Futbol Total DIRECTV	@DTVTotal	415, 6 K	Programa deportivo
70	GOLPERU	@GOLPERUoficial	78,1 K	Programa deportivo
71	GOLTV	@GolTV	65,7 K	Programa deportivo
72	Latina Deportes	@LatinaDeportes	22,6 K	Programa deportivo
73	Movistar Deportes	@MovistarDeporPe	122,4 K	Programa deportivo
74	PBO Digital	@PBODigital	419, 3 K	Programa deportivo
75	RPP Deportes	@RPPDeportes	489,1 K	Programa deportivo
76	Teledeportes	@TeledeportesTV5	17,4 K	Programa deportivo
77	TVPeru Deportes	@TVPeruDeportes	24,3 K	Programa deportivo
78	La Banda del Chino	@bandadelchino	70 K	Programa televisivo
79	Prom Perú	@promperu	54,1 K	Promoción Perú
80	Selección Peruana de Futbol	@SeleccionPeru	483,5 K	Selección peruana

Cuentas de Usuario Twitter

d. Instalación de la API GetOldTweets al sistema de programación.

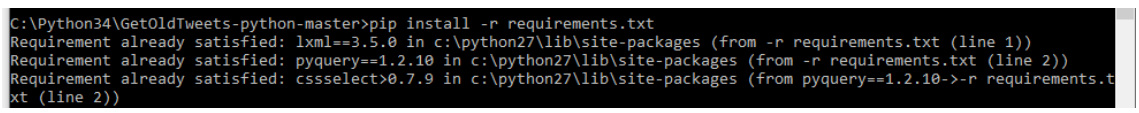
Paso 1: Descargar la API GetOldTweets-python de (Jefferson, 2016)

Paso 2: Copiar la API dentro de la carpeta que contiene la instalación de Python 3.4 del sistema.



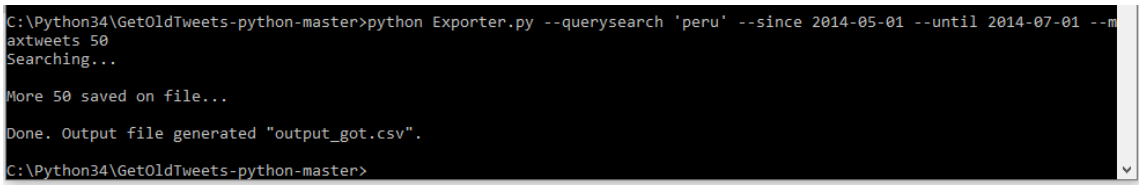
Paso 3: Ejecutar el comando PIP desde la línea de comandos para terminar de descargar las dependencias necesarias (xml y pyquery).

>pip install –r requirements.txt (enter)



Paso 4: Probar funcionamiento, ejecute:

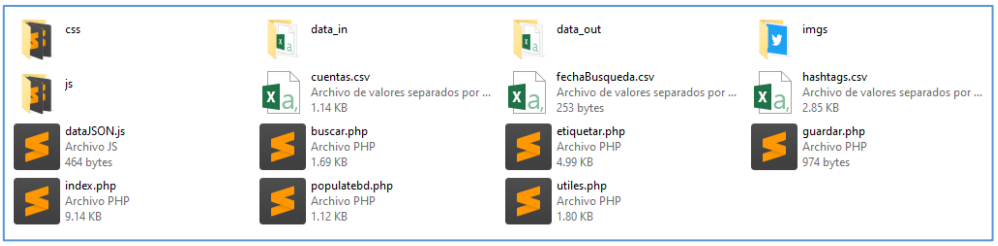
>python Exporter.py –querysearch ‘peru’ –since 2014-05-01 –until 2014-07-01 –maxtweets 50 (enter)



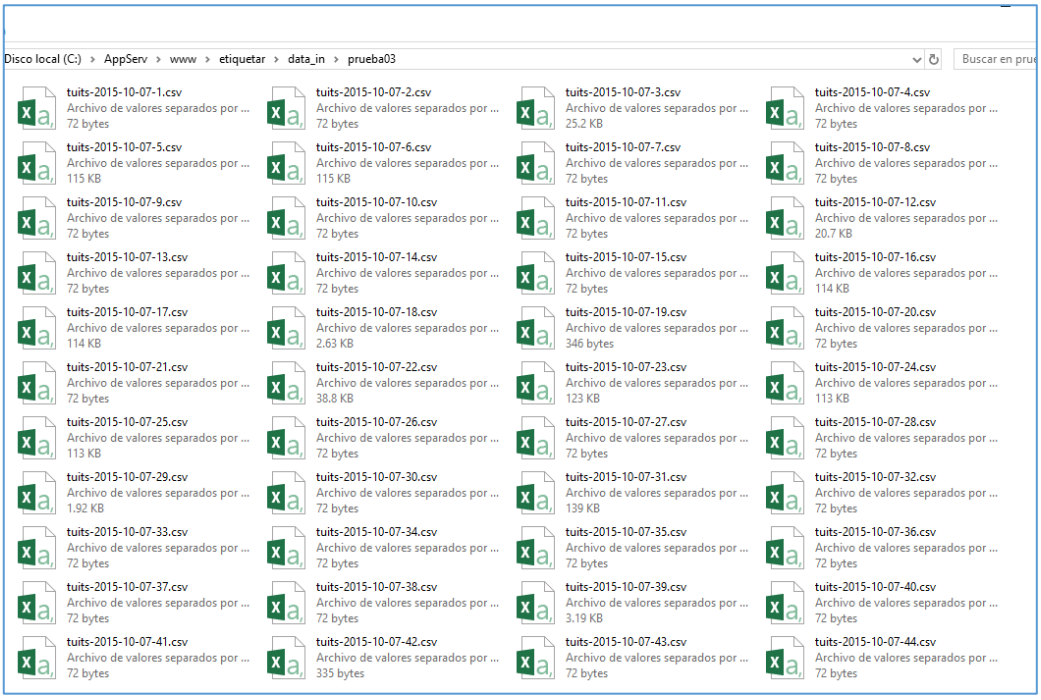
**Exporter.py es un script que utiliza la API GetOldTweets predeterminado para probar el correcto funcionamiento de la API.

Anexo 2. Funcionamiento de la App Etiquetador Manual.

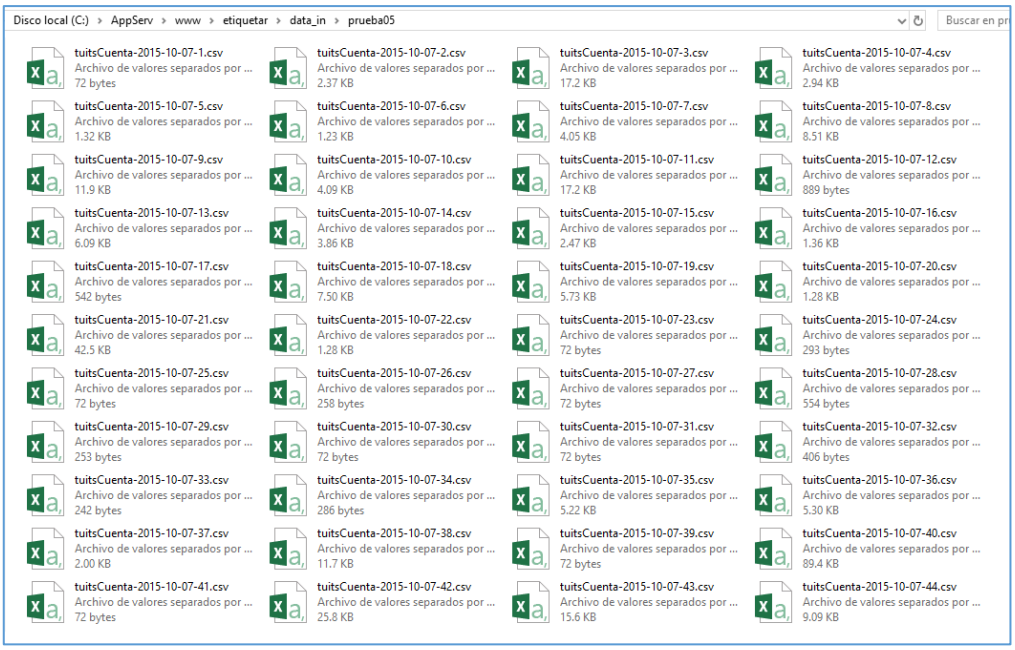
a. Partes de la aplicación Web:



La carpeta **data_in/prueba03**, contiene 1784 archivos CSV arrastrados y que contienen tuits en estado RAW DATA correspondientes a los 11 intervalos de fecha y 162 hashtags. Han sido codificados de la siguiente manera: tuit + fecha match FIFA + Nro de Hashtag:



La carpeta **data_in/prueba05**, contiene 880 archivos CSV arrastrados y que contienen tuits en estado RAW DATA correspondientes a los 11 intervalos de fecha y 80 cuentas de usuario. Han sido codificados de la siguiente manera: tuitCuenta + fecha match FIFA + Nro de Cuenta usuario:



b. Base de Datos; está compuesta por 5 tablas: fechas, hashtags, peruarusia2018, tuits y usuarios.

La tabla **tuits** recoge cada tuit que ha sido etiquetado manualmente, resguardando su origen desde el RAW DATA, la columna **id** sigue el siguiente estándar para lograrlo:

- Primer digito: 1 si el tuit recolectado proviene por búsqueda fecha y hashtag, y 3 si el tuit recolectado proviene por búsqueda fecha y cuenta de usuario.
- Del 2do al 9no dígito: fecha match de publicación del tuit concatenado=Año (4 dígitos) + Mes (2 dígitos) + Día (02 dígitos); se completa con 0 por delante para mantener la cantidad de dígitos ocupados.
- Del 10mo al 12avo dígito: Número del hashtag o número de cuenta. Se mantiene los ceros por delante para completar 3 dígitos.
- Del 13avo al 15avo dígito: Número de orden del tuit que ha sido recolectado dentro de un archivo. Por ejemplo el siguiente ID se interpreta del siguiente modo:

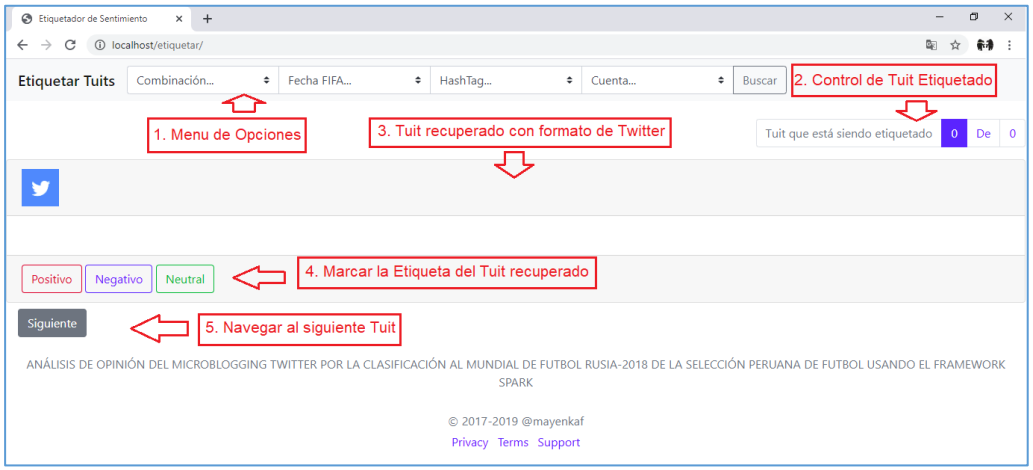
El tuit etiquetado proviene de la carpeta data_in/prueba03, cuyo nombre es tuits-2015-10-09-97.csv, el hashtag es “#Perdimoscomosiempre” y el tuit ocupa la línea 21.

id_tuit
120151009097021

De esa forma se resguarda el origen del tuit en estado RAW DATA desde el dataset.

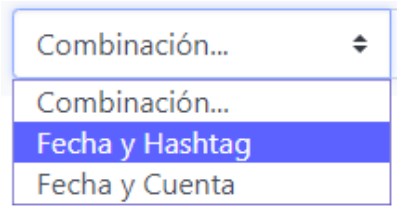
c. Manual de usuario para etiquetar cada tuit manualmente:

1. Al Ingresar a la App, se tiene acceso: Menú de opciones, control del tuit que está siendo etiquetado, los datos de un tuit recuperado, botones para marcar la etiqueta del tuit, y el botón para navegar al siguiente Tuit.

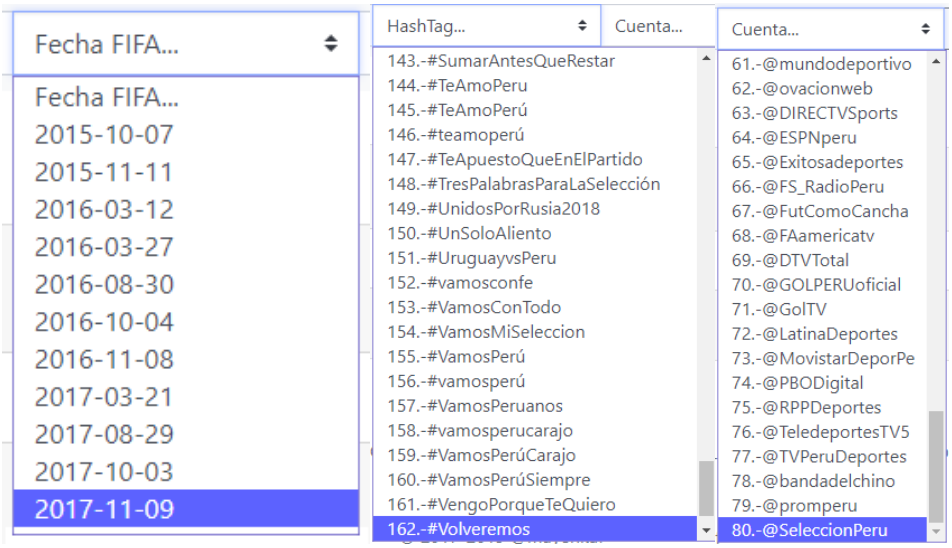


2. El Menú Opciones, le permite a través del botón **Buscar** recuperar todos los tuits que se encuentran en una archivo CSV, esta son las siguientes opciones de búsqueda disponibles:

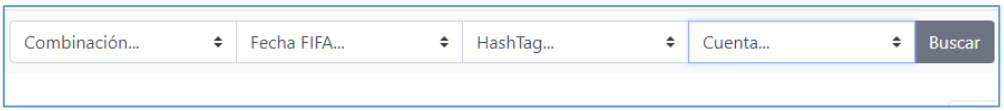
a. Elegir la Combinación de arrastre del Tuit



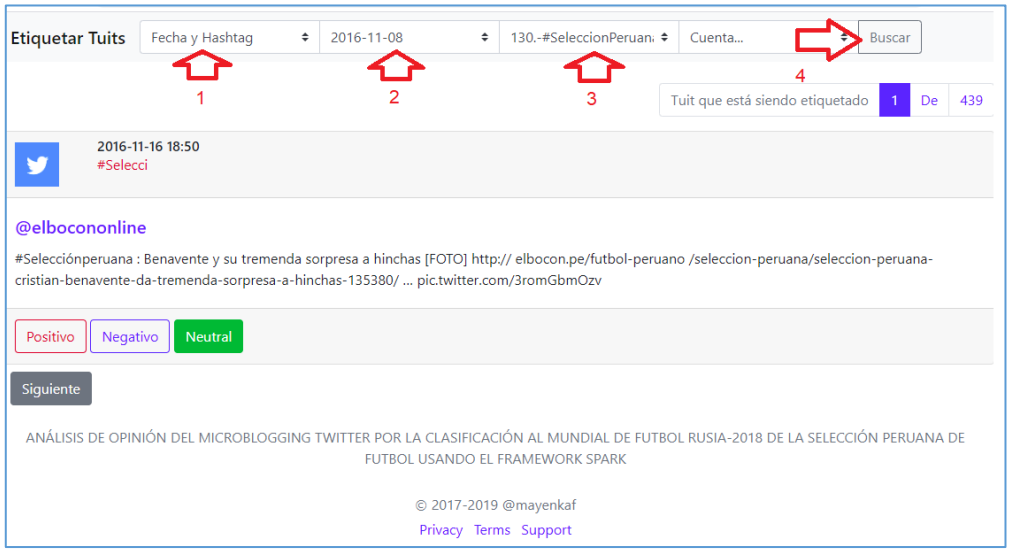
b. Elegir una Fecha FIFA (disponible 11 intervalos de fechas FIFA), Hashtag (disponible 162) o cuenta de usuario (disponible 80)



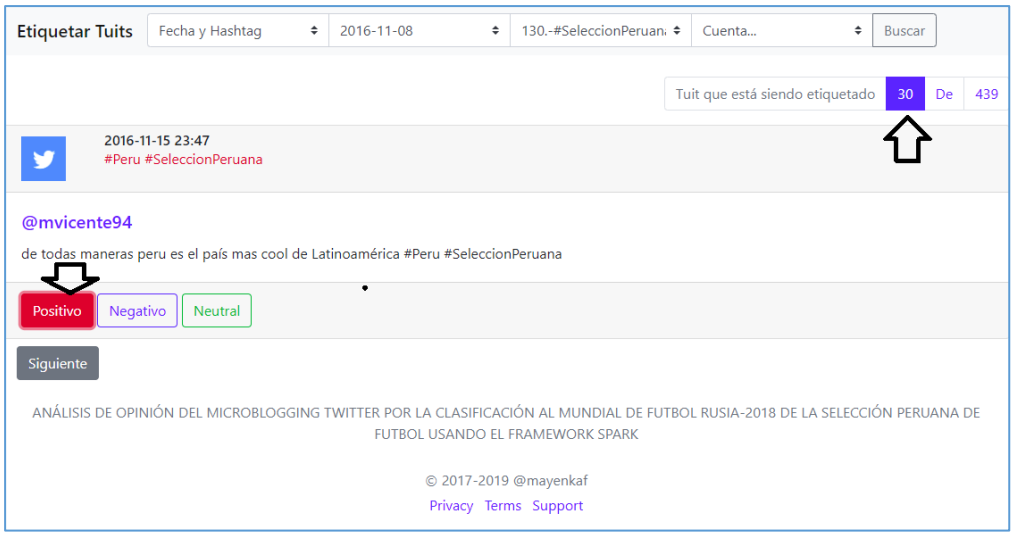
c. Botón Buscar:

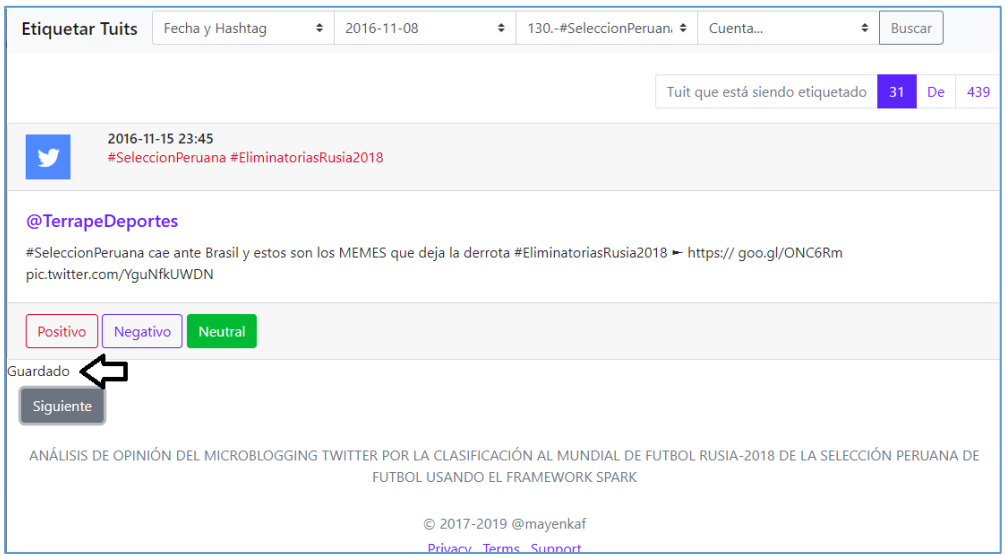


d. Por ejemplo: Para recuperar los tuits del Match del 2016-11-08, con el hashtag #SeleccionPeruana, se deberá elegir: Combinación = Fecha y Hashtag, Fecha = 2016-11-08, Hashtag = 130.#SeleccionPeruana, y el Botón Buscar.



e. Una vez que haya leído le tuit, lo podrá etiquetar con los botones Positivo o Negativo, y si cree que no guarda sentimientos no es necesario hacer nada ya que cada tuit se supone es Neutral, para pasar al siguiente tuit, tendrá que cliquear en el Botón Siguiente. Si etiqueta un tuit con Positivo, entonces elegirá el Botón Positivo y al cliquear en el botón Siguiente se guardará en la base de datos como un tuit con etiqueta POSITIVA.





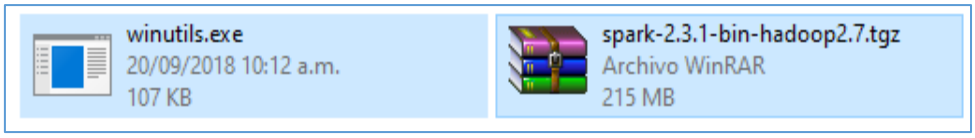
3. De esta forma se acumulan solo los tuits etiquetados como positivos o negativos, más no los neutrales.
4. Observe también que el contador de tuits controla el número de tuit que está siendo recuperado, leído y posiblemente etiquetado.
5. En la tabla **tuits,** se ha almacenado del siguiente modo:

polaridad	id_tuit	fecha	consulta	usuario	texto
4	120161115130030	2016-11-15 23:47	#SeleccionPeruana	@mvicente94	de todas maneras peru es el país mas cool de Latin...

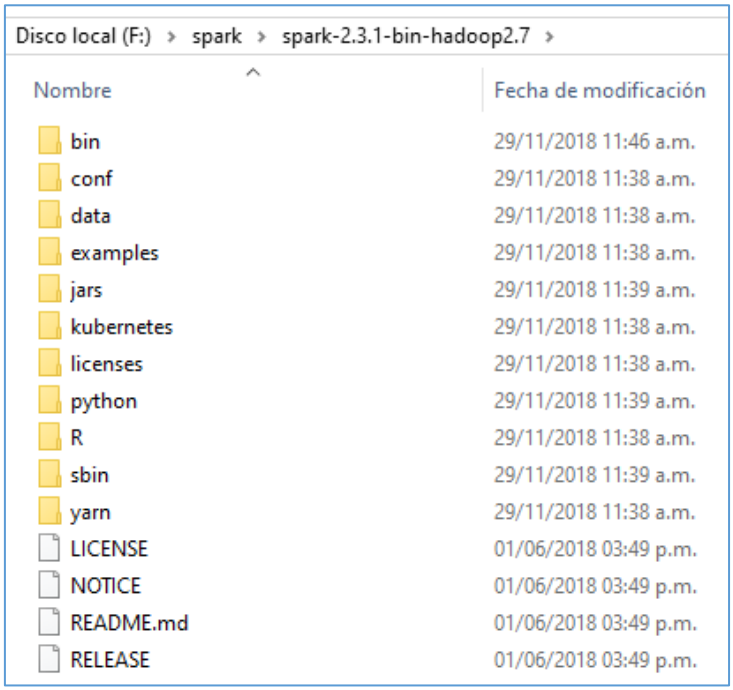
Anexo 3. Instalación de Spark y Anaconda.

a. Instalar Spark en Windows 10

- Prerequisitos: tener instalado: Java SDK 1.8, Scala 2.11, Python 3.4.
- Descargar de https://spark.apache.org/downloads.html Spark release y Hadoop empaquetado 2.3.1.
- Descargar el archivo winutils.exe de: https://github.com/steveloughran/winutils/blob/master/hadoop-2.7.1/bin/winutils.exe



- Descomprimir el Zip de Spark-Hadoop y copiarlos a una carpeta de nombre Spark en una unidad del disco duro.

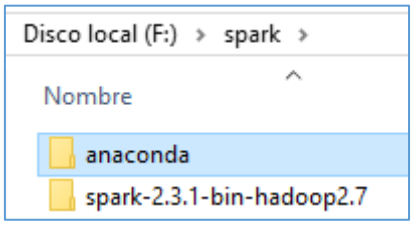


- Copiar el archivo **winutils.exe** dentro de la carpeta **bin.**
- Añadir rutas de acceso a la carpeta **bin** a usuario Windows Administrador: HADOOP_HOME, JAVA_HOME, SCALA_HOME, SPARK_HOME.
- Comprobar funcionamiento, ejecute >**spark-shell** y podrá usar Spark en el lenguaje Scala:

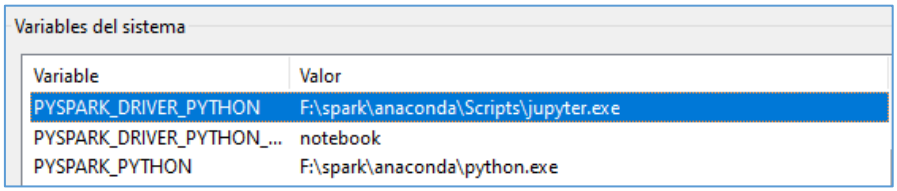
```
F:\spark\spark-2.3.1-bin-hadoop2.7\bin>spark-shell
2020-01-03 11:46:22 WARN  NativeCodeLoader:62 - Unable to load native-hadoop
java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogL
Spark context Web UI available at http://MAYENKA:4040
Spark context available as 'sc' (master = local[*], app id = local-1578070026
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.3.1
      /_/

Using Scala version 2.11.8 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_92)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

b. Instalar Anaconda en Windows 10

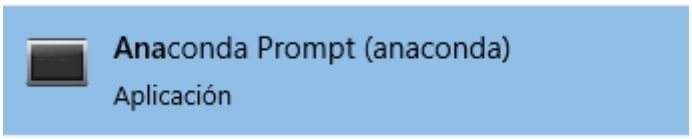
- Descargue anaconda Python 3.7 desde: https://www.anaconda.com/distribution/
- Siga los pasos de instalación por defecto, solo cambie el paso de ubicación de copiado de los archivos, seleccione la misma carpeta donde instaló Spark. Al final de la instalación deberá tener los siguiente:



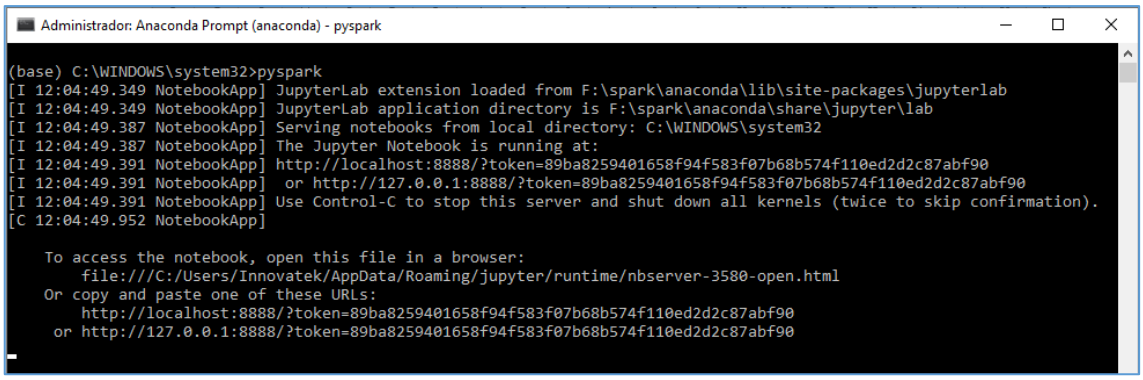
- Añada rutas a las variables del sistema:

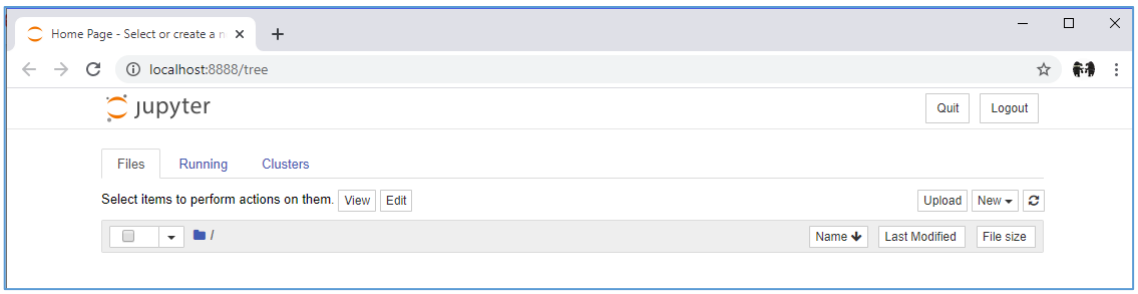


- Para comprobar el funcionamiento, realice lo siguiente:
  - Ejecute el Shell de Anaconda como Administrador



  - Dentro del Shell escriba **pyspark** y espere hasta que se despliegue el servidor Notebook Jupyter:





- Cree un nuevo Notebook, y programará interactivamente con Spark en el lenguaje Python, o pySpark.

Anexo 4. URLs de Recursos Disponibles de esta Tesis.

Recurso	URL
Artículos fuente traducidos de esta tesis	http://bit.ly/2TnGroG
Tuits en Raw Data – CSV	http://bit.ly/2QSKBU2
PeruARusia2018.csv	http://bit.ly/2Rf78cr
Código del Modelo de Análisis de Sentimiento en Spark	https://github.com/mayenkaf/AnalisisOpinionconSpark.git