Informe Técnico - Technical Report
DPTOIA-IT-TFM
julio, 2019

Minería de Opiniones para el enriquecimiento continuo de Curricula.

Autor: Edgar Naranjo Fuentes
Tutora: Vivian Félix López Batista
Tutor: Juan José San Martín

Departamento de Informática y Automática
Universidad de Salamanca

Modelo de Declaración de Autoría para el Trabajo de Fin de Máster

Declaro que he redactado el Trabajo de Fin de Máster (TFM) titulado **Minería de Opiniones para el enriquecimiento continuo de Curricula** del Máster Universitario en Ingeniería Informática de la **Universidad de Salamanca** en el segundo semestre del curso académico **2018-2019** de forma autónoma, con la ayuda de las fuentes y la literatura citadas en la bibliografía, y que he identificado como tales todas las partes tomadas de las fuentes y de la literatura indicada, textualmente o conforme a su sentido.

En Salamanca, 28 de junio de 2019.
Fdo.: Edgar Naranjo Fuentes

Facultad de Ciencias. Plaza de los Caídos, s/n, 37008. Salamanca
Tel.: +34 923 29 44 00 ext.: 1513 Fax:+34 923 29 45 14
http://masterii.usal.es
masterii@usal.es

Mientras los filósofos discuten si es
posible o no la inteligencia artificial,
los investigadores la construyen

C. Frabetti

Resumen

Tras el cambio de paradigma producido por la Web 2.0 en el que los usuarios dejan de ser meros sujetos pasivos y se convierten en los protagonistas, el volumen de información en internet ha aumentado exponencialmente.

En comparación con décadas anteriores, la participación activa de los usuarios en las redes sociales, posibilita que los datos en la red sean una valiosa fuente de información de cara al análisis de datos. Análogamente, muchas webs permiten hoy día que los usuarios expresen su sentir acerca de cualquier producto o servicio recibido, hecho que está siendo utilizado por las empresas en la toma de decisiones de cara a mejorar sus políticas de marketing.

La recolección de esta información subjetiva es una parte crítica desde la perspectiva del procesamiento de textos y el lenguaje natural, para lograr la minería de opiniones.

Entre los diferentes métodos de análisis de sentimientos valorados, el enfoque elegido para este trabajo es el método de aprendizaje automático, basándose en una clasificación no supervisada para el entrenamiento de sus clasificadores; teniendo como ventaja la facilidad de adaptación a contextos específicos, a pesar del alto coste de preparación que requieren estos tipos de datos.

La idea subyacente es el uso de técnicas de minería de datos y procesamiento de lenguaje natural para obtener de forma automática, conocimiento útil acerca de las opiniones, preferencias y tendencias de los usuarios.

Para finalmente haciendo uso de redes neuronales artificiales, concretamente los Mapas Auto-organizados entrenar una red neuronal que sea capaz de en función de lo expresado por el usuario, discernir su estado de ánimo, gustos, experiencias para así ayudar a la Empresa de Selección de Personal (Faster Empleo ETT) a la hora de satisfacer las necesidades de sus clientes y empleados.

Palabras Claves:

Web 2.0, información subjetiva, sentimientos, procesamiento de textos, lenguaje natural, aprendizaje automático, minería de datos, redes neuronales artificiales

Abstract

After the paradigm shift produced by the Web 2.0 in which users stop being mere passive subjects and become the protagonists, the volume of information on the internet has increased exponentially.

Compared with previous decades, the active participation of users in social networks makes it possible for data in the network to be a valuable source of information for data analysis. Analogously, many websites now allow users to express their feelings about any product or service received, a fact that is being used by companies in decision making in order to improve their marketing policies.

The collection of this information is a part from the perspective of texts and the natural language processing to achieve the mining of opinions.

Among the different methods of analyzing valued feelings, the approach chosen for this work is the machine learning method, based on an unsupervised classification for the training of their classifiers; having as an advantage the ease of adaptation to specific contexts, despite the high preparation cost required by these types of data.

The underlying idea is the use of data mining techniques and natural language processing to obtain automatically useful knowledge about the opinions, preferences and trends of users.

To finally making use of artificial neural networks, specifically Self-organized Maps train a neural network that is capable of depending on what is expressed by the user discern their mood, tastes and experiences in order to help the Personnel Selection Company (Faster Employment ETT) to meet the needs of its customers and employees.

Keywords:

Web 2.0, subjective information, feelings, word processing, natural language, machine learning, data mining, artificial neural networks.

Índice

Índice de figuras

Indice de tablas

Introducción

El creciente desarrollo de la tecnología que vivimos en las últimas décadas se debe en buena parte a los increíbles avances que se han producido en los campos de la **informática y la comunicación**; como lo es la universalización de las nuevas tecnologías, así como la generalización de las redes de comunicación y la globalización de la información.

Desde acceder a masivas cantidades de información en Internet, las relaciones entre las personas, la manera cómo organizamos y dirigimos los procesos, a simplemente realizar una compra online; la **tecnología informática** continúa mejorando nuestra calidad de vida tanto a nivel laboral como personal.

Tras el cambio de paradigma producido por la **Web 2.0** en el que los usuarios dejan de ser meros sujetos pasivos y se convierten en los protagonistas, el volumen de información en internet ha aumentado exponencialmente [1].

En comparación con décadas anteriores, la participación activa de los usuarios en las redes sociales, compartir contenidos, gustos, preferencias, experiencias y/o conocimientos; posibilita que los datos en la red sean una valiosa fuente de información de cara al análisis de datos. Análogamente, muchas webs permiten hoy día que los usuarios expresen su sentir acerca de cualquier producto o servicio recibido.

La recolección de esta información subjetiva es una parte crítica desde la perspectiva del **procesamiento de textos** y el **lenguaje natural**.

Anteriormente la retroalimentación o la manera de saber cómo los clientes se sentían acerca de un servicio o todo lo relacionado a un producto, era mediante test, encuestas o una investigación de mercados. Lo cual ha cambiado con el nuevo paradigma ya que la información es asequible a todos, teniendo en cuenta que los usuarios dan su criterio acerca de sus necesidades, nivel de satisfacción, opiniones de servicios o productos, preferencias políticas, gustos, etc.

Muchas áreas han tenido que reinventarse para lograr mantenerse en el mercado y conseguir visibilidad. En ese sentido, hoy en día para realizar un estudio de mercado objetivo puede efectuarse de manera muy efectiva, a través de las redes sociales. Aprovechando así los datos proporcionados por los usuarios, tratándolos y transformándolos en información que puede ser utilizada para agregar valor a sus servicios.

El inconveniente ya no es entonces la inmensa cantidad de información que existe, sino cómo extraer las opiniones y sentimientos expresados para ser analizados.

El reto de la interpretación de las opiniones y sentimientos expresados en forma textual dio lugar a la aparición de **análisis de los sentimientos** o **minería de opiniones**, que se remite al procesamiento del lenguaje natural, análisis de texto y a la lingüística computacional, para identificar y extraer información subjetiva en el texto, con el fin de que se pueda clasificar como positiva o negativa. Lo anteriormente expresado, posibilita hacer una encuesta de gran alcance, no invasiva, rápida, auténtica, barata y automática [2].

Existen un sin número de técnicas, algoritmos, plataformas, etc. [18] que se utilizan para el análisis y procesamiento de textos, aunque lo novedoso ha sido en ¿cómo adaptar estos algoritmos para procesar micro textos como es el caso de **Twitter** y de otras redes sociales?

En el caso particular de Twitter a la hora de redactar sus mensajes, se debe tener en cuenta el uso de emoticonos, abreviaturas, repetición de letras, frases, palabras tanto en inglés como español, además de construcciones específicas de la propia red como es el caso de los *hashtags*; lo cual requiere un trabajo minucioso en el procesado de textos con *Natural Language Processing* (NLP).

La propuesta de este trabajo es la combinación de **métodos de análisis de sentimientos** extraídos a través de mensajes escritos en la red social Twitter; con *Machine Learning* (ML)[1], específicamente **redes neuronales artificiales** aprovechando la reciente atención que ha captado en los profesionales dedicados a la estadística y al análisis de datos.

Convirtiendo así a través de técnicas de **Minería de Datos** en información valiosa el conjunto de datos recolectados en dicha red social, de forma tal que la Empresa de Selección de Personal (Faster Empleo ETT) pueda encaminar acciones para satisfacer las necesidades tanto de sus clientes como empleados. Proporcionándoles a sus cliente, empleados cualificados, de acuerdo a los requisitos previos establecidos y al empleado un puesto de trabajo que satisfaga sus necesidades, aspiraciones y mejore su calidad de vida.

Objetivos

Durante mucho tiempo el procesamiento de información se centró en datos que no resultaban del todo útil en la toma de decisiones, y por lo general no se tenía en cuenta las opiniones ni los sentimientos de los usuarios en internet. Sin embargo como señala [3], la información sin sentimiento es incompleta.

El principal inconveniente para generar información valiosa y útil está en la comprensión del lenguaje y en la estructuración de la **información semántica** obtenida; de ahí que uno de nuestros objetivos sea analizar y comparar sistemas de minería de opiniones basados en redes sociales, a partir de técnicas de procesamiento del lenguaje natural y minería de datos. Concretamente investigar la utilización de redes neuronales, a través de los Mapas Auto-organizados, ya que estos son una poderosa herramienta para la clasificación no supervisada, con muy buenas prestaciones de visualización de resultados y que toman en cuenta el contenido semántico de los datos.

Se desea extraer información subjetiva como opiniones, preferencias, sentimientos y emociones de los usuarios; estructurarla, procesarla y clasificarla para obtener información útil.

[1] *Machine Learning - Aprendizaje Automático: Rama de la* ***Inteligencia Artificial*** *cuyo objetivo es desarrollar técnicas que permitan a las computadoras aprender. Generaliza comportamientos a partir de una información suministrada.*

Se pretende además identificar sentimientos expresados mediante mensajes escritos en Twitter, para una vez clasificados, relacionar los perfiles de opinión con los usuarios del sistema y poder tomar acciones comerciales que ayuden en la mejora continua de la empresa.

Posteriormente representar gráficamente dichos resultados a través de una interfaz intuitiva, ágil y fácil de manejar para los usuarios finales.

El usuario podrá además interactuar, consultar o descargar parte o la totalidad de los datos; así como disponer de estadísticas de los mismos.

Organización del Trabajo

El presente trabajo se encuentra estructurado de la siguiente forma: en el Estado del Arte se elabora una revisión acerca de técnicas, aplicaciones, métodos y trabajos existentes relacionados con NLP para el análisis de sentimientos; en la sección Metodología, se describe el análisis del problema, las herramientas utilizadas, las pruebas realizadas y la implementación del sistema; en la sección Prototipo de la herramienta, se detallan las interacciones con el sistema y las representaciones gráficas de los resultados obtenidos. Por último, en la sección Conclusiones se sintetizan las conclusiones sobre el trabajo y las líneas futuras de investigación.

Estado del Arte

En esta sección se pretende en un primer momento realizar el estado del arte acerca del análisis de sentimientos en redes sociales como Twitter, partiendo de las definiciones de NLP y aprendizaje automático. Se presentan las técnicas más utilizadas en clasificación subjetiva de textos y en la evaluación de clasificadores. Se incluye además, un análisis sobre algunas soluciones existentes asociadas al tema.

Procesamiento del Lenguaje Natural

El lenguaje natural se utiliza a diario como medio de comunicación entre humanos, es nuestra forma de comunicarnos por excelencia. Aunque a simple vista sea un gesto sencillo, casi inconsciente; es un proceso que implica millones de conexiones neuronales y complejos procesos corporales de captación, comprensión, trasmisión de conocimientos, sentimientos y emociones.

Según avanzaba la tecnología, la cantidad de información generada en forma de lenguaje natural, crecía también de manera vertiginosa; comenzando así una era marcada por un volumen inmenso de información.

Para que dicha información se transforme en conocimiento útil, debe de ser procesada de forma tal que se puedan hacer deducciones lógicas respecto a su contenido, generalizar o especificar, resumir información, etc.

A raíz de la necesidad de entender y procesar ese volumen de información, surge en los años **60** una rama de la informática, llamada procesamiento del lenguaje natural en inglés ***Natural Language Processing*** [18], que busca precisamente permitir la interacción y comunicación entre personas y máquinas, a partir del estudio de los problemas originados de la generación, comprensión y procesamiento automático del lenguaje natural.

Uno de los campos de acción en los que se desarrolla dicha ciencia es en cómo el lenguaje puede usarse para llevar a cabo diferentes tareas, dígase transformación de textos, sistemas de diálogo interactivos, traducciones automáticas, etc.; y cómo modelar el conocimiento.

Sin embargo, el NLP plantea muchos problemas: los múltiples significados de cada palabra, los acentos de cada zona, la jerga de cada lugar, expresiones típicas, lenguaje ambiguo, ironías, etc. Para un ordenador, analizar una frase de diez palabras consiste en un número enorme de posibilidades. No sólo tiene que mirar esas diez palabras, sino sus posibles significados, y como cada una de ellas se relaciona con las demás.

Estas propiedades que reducen en buena medida la efectividad de los sistemas de recuperación de información de textos, son conocidas como **variación** y **ambigüedad lingüística**. Que consisten en el uso de diferentes palabras o expresiones para manifestar una misma idea, y por otro lado, la ambigüedad lingüística ocurre cuando una palabra o frase presenta un abanico de posibilidades de interpretación.

Ambos fenómenos inciden en el proceso de recuperación de información, aunque de manera distinta, una a través del silencio y otra a través del ruido documental. La variación lingüística ocasiona el silencio documental, que resulta en la omisión de documentos relevantes o pertinentes que cubren la necesidad de información por el uso de términos no adecuados o demasiado específicos para definir la búsqueda. En cambio, la ambigüedad lingüística provoca el ruido documental, que incorpora documentos no significativos a los resultados de la búsqueda debido al uso de palabras claves genéricas, lo que conduce a la recuperación de términos con significados diferentes al requerido [4].

Estas dos características dificultan considerablemente el tratamiento automatizado del lenguaje.

A continuación, vemos algunos de los componentes de los sistemas que usan NLP, ya que no todos se aplican en cualquier tarea, sino que depende del objetivo de la aplicación, requiriendo para ello la realización de una serie de procesos estructurados en cuatro niveles de análisis: morfológico, sintáctico, semántico y pragmático.

- **Análisis morfológico o léxico**

  Consiste en el análisis interno de las palabras que forman oraciones para extraer lemas, rasgos flexivos, unidades léxicas compuestas. Es esencial extraer para la información básica: categoría sintáctica y significado léxico.

- **Análisis sintáctico**

  Consiste en el análisis de la estructura de las oraciones de acuerdo con el modelo gramatical empleado (lógico o estadístico).

- **Análisis semántico**

  Proporciona la interpretación de las oraciones, una vez eliminadas las ambigüedades morfosintácticas.

- **Análisis pragmático**

  Incorpora el análisis del contexto de uso a la interpretación final. Aquí se incluye el tratamiento del lenguaje figurado (metáfora e ironía) como el conocimiento del mundo específico necesario para entender un texto especializado.

Historia

En algunas fuentes bibliográficas cuando hablan del origen o nacimiento del **procesamiento de lenguaje natural** hacen referencia al período bélico de la Segunda Guerra Mundial, en torno de los años **1940**.

Aunque a ciencia cierta podría decirse que nació en la década de **1960**, como un subárea de la **Inteligencia Artificial y la Lingüística**, con el objetivo de estudiar los problemas derivados de la generación y comprensión automática del lenguaje natural. La traducción automática, por ejemplo, ya había nacido a finales de la década de los cuarenta, antes de que se acuñara la propia expresión **Inteligencia Artificial**.

En sus orígenes, sus métodos tuvieron gran aceptación y éxito, no obstante, cuando sus aplicaciones fueron llevadas a la práctica, en entornos no controlados y con vocabularios genéricos, empezaron a surgir multitud de dificultades. Entre ellas, pueden mencionarse por ejemplo los problemas de polisemia[2] y sinonimia[3].

Entre algunos de los avances, se encuentra el experimento de *Georgetown* en **1954**, con el cual se logró la traducción automática de más de sesenta frases del ruso al inglés y a pesar de haber sido un experimento de pequeña escala, con apenas 250 palabras, contribuyó en gran medida a elevar las expectativas de los sistemas automáticos capaces de realizar traducción de alta calidad en un futuro próximo [5].

Los primeros experimentos basados en la sustitución de palabra por palabra produjeron resultados rudimentarios, causado por diversos motivos como la escasa potencia de las máquinas de aquella época, el desarrollo muy pobre de las interfaces lingüísticas y la carencia de un lenguaje de programación de alto nivel. Sumado a eso, muchos idiomas no tenían una formalización del lenguaje con lo cual su representación sintáctica y semántica se hacía muy compleja [6].

Los finales de los años 60 estuvieron influenciados por la **Inteligencia Artificial**,

[2] *Palabra que tiene varios significados.*
[3] *Relación semántica de identidad o semejanza de significados.*

sobresaliendo los sistemas de preguntas y respuestas desarrollados por el *Massachusetts Institute of Technology* (MIT), uno de los grupos pioneros en el ámbito [18].

Entre **1964** y **1967** nace *Eliza*, capaz de conversar con una persona y reproducir las habilidades de conversación de un psicólogo. Procesaba las palabras en el lenguaje natural y contestaba de manera coherente con frases programadas adecuadas o respondía con una frase para que el individuo siguiera hablando. Sustentada en los principios de la psicología de Carl Rogers, trataba todavía de crear una empatía con el individuo para que él se sintiera escuchado [8].

Para los años **1968-70**, Terry Winograd había desarrollado *SHRDLU* en el MIT; un programa que realizaba un diálogo simple (mediante teletipo) con un usuario, sobre un pequeño mundo de objetos (el mundo BLOCKS) que se muestra en una pantalla de visualización (DEC-340 conectada a una computadora PDP-6) [9]. Dicho programa respondía preguntas ejecutando comandos y aceptando información en un diálogo interactivo en inglés. Algo un tanto difícil, ya que la comprensión del inglés requiere un estudio integrado de la semántica y la inferencia de la sintaxis.

En esta época se profundiza además en la separación de procesamiento (parsers) y la descripción del conocimiento lingüístico y en la explicitación de nivel de representación semántica. Se percibe la necesidad de utilizar conocimiento sobre el mundo (*proyecto CYC, Lenat*[4]) y se trabaja en la traducción automática en dominios limitados, como por ejemplo la meteorología.

La década de los años **80**, fue un período más ambicioso. Se incrementó la confianza y consolidación de los conocimientos, así como una expansión de la comunidad a través de más recursos prácticos, herramientas disponibles y sistemas comerciales [10].

Fue estimulado el uso de la teoría de las gramáticas formales orientadas a un tratamiento computacional en lugar de la teoría lingüística y por el uso de la lógica para la representación del conocimiento en la Inteligencia Artificial [10]. Dicha transformación se pudo llevar a cabo a causa del aumento constante del poder de cómputo de los ordenadores, lo que permitía aumentar las investigaciones y desarrollar más programas volcados al NLP, permitiendo así que los primeros sistemas de traducción automática estadística se desarrollaron en esta fase [11].

Para los años **90** se había iniciado una nueva fase donde el enfoque estadístico se hizo cada vez más influyente. La disponibilidad de grandes cantidades de texto extraídos de la web reorienta el área. Surgen los primeros resultados robustos con métodos probabilísticos y comienza a utilizarse el aprendizaje automático.

En este período se destacaron las herramientas de extracción de información y de resúmenes automáticos, las cuales tuvieron la tarea de gestionar el enorme volumen de información electrónica disponible en Internet. Surgieron además los modelos ocultos de Markov (*Hidden Markov Modelling* o HMM) logrando así alcanzar un avance rápido y natural en las transcripciones del reconocimiento de voz [10].

[4] *Proyecto que intenta ensamblar una ontología comprensiva y una base de datos de conocimiento general con el fin de permitir a las aplicaciones de inteligencia artificial realizar razonamientos del tipo humano.*

Para el año **2000**, ya se hablaba del NLP como la tendencia para la siguiente década. Se hace énfasis en la semántica y representación del conocimiento, en la integración de técnicas simbólicas y probabilísticas. Se evidencia además mayor integración de componentes de lenguaje natural en otros sistemas.

Surgieron entonces varios intentos de mejoras, como los modelos bayesianos que intentan mejorar los modelos clásicos, basados en la proximidad entre espacios vectoriales. Recientemente, se ha venido trabajando en mejorar los enfoques probabilísticos más clásicos, como el bayesiano [12].

El procesamiento de lenguaje natural sigue creciendo; los métodos, los usos, las oportunidades que brinda y su estrecha relación con muchos de los avances tecnológicos que se avecinan lo convierten en un conocimiento importante en el mundo de los grandes volúmenes de información.

Técnicas de NLP

Antes de comenzar a trabajar con toda la información obtenida, es importante preparar los datos, ya que estos pueden estar impuros, conducir a la extracción de patrones/reglas pocos útiles, contener inconsistencias (incluyendo discrepancias). De ahí que el primer paso hacia la clasificación de textos, sea el preprocesamiento[5].

A pesar de que el preprocesamiento permite aplicar los modelos de Minería de Datos de forma más rápida y sencilla, obteniendo modelos o patrones de más calidad: precisión e/o interpretabilidad. Tiene como inconveniente que no es un área totalmente estructurada con una metodología concreta de actuación para todos los problemas. Cada problema puede requerir una actuación o tipo de herramienta diferente.

Algunas de las tareas de **preprocesamiento** fundamentales para la eliminación de atributos irrelevantes y mejorar así la eficiencia del proceso de **Minería de Datos** se listan a continuación:

- **Normalización**

  Es una técnica que se aplica a un conjunto de datos para reducir su redundancia, con el objetivo de homogenizar todo el texto para que se pueda alcanzar una mejor efectividad del clasificador eliminando términos que no aportan información a la tarea de procesamiento. En el presente trabajo se utilizan técnicas como la capitalización (pasando todo a minúscula); la eliminación de los signos de puntuación, caracteres especiales, fechas, números y palabras con dos letras o menos.

- **Tokenización**

  Consiste en la separación de una cadena de caracteres en pequeños fragmentos (*tokens*). A su vez un token es una parte del todo; eso significa que una palabra es un token de una oración y una oración es un token de un párrafo.

[5] *Engloba a todas aquellas técnicas de análisis de datos que permite mejorar la calidad de un conjunto de datos.*

- **Palabras vacías (*Stopwords*)**

  Nombre que reciben las palabras o términos sin significado como artículos, pronombres, preposiciones, etc. que son filtradas antes del procesamiento de datos y que comúnmente no contribuyen para el significado de una oración, por lo menos en el contexto de recuperación de la información y del NLP. Ignora además el orden o la categoría gramatical de los términos y se centra en su frecuencia.

- **Etiquetado gramatical (*PoS Tagging*)**

  Se denomina con este término el proceso de etiquetado gramatical. En concreto, consiste en identificar a qué categoría (nombre, adjetivo, verbo...) gramatical pertenece cada palabra de un texto. Este proceso se puede realizar de acuerdo con la definición de la palabra o el contexto en que aparece.

  Uno de los usos de este etiquetado tiene lugar en el contexto de la lingüística computacional, mediante el empleo de algoritmos que realizan el etiquetado mediante etiquetas descriptivas predefinidas.

Aplicaciones del NLP

En los últimos años, las aportaciones que se han hecho desde este dominio han mejorado sustancialmente, permitiendo el procesamiento de ingentes cantidades de información en formato texto con un grado de eficacia aceptable. Muestra de ello es la aplicación de estas técnicas como un componente esencial en los motores de búsqueda web, en las herramientas de traducción automática, en la generación automática de resúmenes, tratamiento automático de texto [18], etc.

Destacándose en este campo algunas áreas como por ejemplo la lingüística, informática, psicología, la biomedicina, y otras como:

- Análisis de sentimiento
- Síntesis del discurso
- Análisis y comprensión del lenguaje
- Reconocimiento del habla
- Síntesis de voz
- Clustering
- Generación de lenguajes naturales
- Traducción automática
- Respuesta a preguntas
- Recuperación de la información

- Extracción de la información
- Resumen automático de textos
- Corrección automática de documentos

Análisis del Sentimiento

El **análisis de sentimiento** o **minería de opiniones** es un tipo de procesamiento del lenguaje natural que se utiliza con el fin de conocer el estado de ánimo u opinión de los usuarios acerca de productos, servicios, marcas, personas y/o instituciones [7]. Se refiere además a los diferentes métodos de lingüística computacional que ayudan a identificar y extraer información subjetiva del contenido existente en el mundo digital (redes sociales, foros, webs, etc.). Gracias al mismo, podemos ser capaces de extraer un valor tangible y directo, como puede ser determinar si un texto extraído de la red Internet contiene connotaciones positivas o negativas.

Es un término que está muy ligado a las redes sociales pero que, en realidad, no está limitado a ellas.

Tanto en las redes sociales y en la red en general se encuentran multitud de textos, en los cuales deben aplicarse subjetividad y no únicamente clasificarlos según su naturaleza o procedencia.

Mediante el análisis del sentimiento, aspiramos a entender, en primer lugar, con qué guarda relación el texto que analizamos; queremos lograr entender cuál es la intención exacta de una frase. Saber si se refiere a una marca, a un producto en concreto o a cualquier otro aspecto.

Estudios realizados en este ámbito ambicionan señalar la polaridad de un texto subjetivo, es decir, determinar si el texto tiene una connotación positiva o negativa, empleando para tal fin diferentes métodos [13], como por ejemplo la detección de la polaridad mediante adjetivos, ya que contienen una faceta emocional y la clasificación automática por medio del uso de textos previamente etiquetados.

Determinar la polaridad de un texto por medio de adjetivos, se centra en preprocesar un texto, buscar en el diccionario las palabras que tengan asignada una polaridad y utilizar la suma de los valores de polaridad para determinar la polaridad global del texto.

Para la resolución de un problema de clasificación de la polaridad se pueden seguir dos estrategias: la **supervisada** y la **no supervisada**. El trabajo que se toma como referencia para los métodos supervisados es **Pang et al.** [14], que utiliza como característica la presencia o no de los términos para el cálculo de la polaridad.

En la estrategia supervisada, además de la utilización de características léxicas, también se ha intentado explotar la información sintáctica. Un ejemplo de ello son los trabajos de **Mullen** y **Collier** [24] y el de **Whitelaw, Garg** y **Argamon** [25], en los que se utilizan los adjetivos para la clasificación de la polaridad.

Por otro lado, los métodos no supervisados se fundamentan en la detección de identificadores de subjetividad u opinión en los textos, para después calcular la polaridad empleando alguna función basada en los indicadores encontrados.

Existen básicamente dos formas de procesar la información obtenida; una es a través del análisis manual y la otra, el análisis de sentimiento automático.

El **análisis manual**, suele darse en casos en los que las palabras claves sobre las que se quiere obtener información pueden representar diferentes significados en diferentes ámbitos, por lo que habrá que estar atento e ir clasificando cada texto en su lugar correspondiente.

Por su parte el **análisis de sentimiento automático**, comienza con el establecimiento de una serie de palabras claves para que cualquier texto que contenga esa palabra o combinación de ellas, quede automáticamente encuadrado en una categoría de una forma previamente definida o descartado directamente.

En comparación con el análisis manual, este último aporta la inmediatez y espontaneidad de la información, la capacidad de procesamiento de datos que, por su alto volumen, variedad y velocidad, se ejecutaría de manera poco eficiente por los humanos.

Web 2.0

El término de **Web 2.0** se refiere al fenómeno social surgido a partir del desarrollo de diversas aplicaciones en Internet, y establece una distinción entre la primera época de la Web (donde el usuario era básicamente un sujeto pasivo que recibía la información o la publicaba, sin que existieran demasiadas posibilidades para que se generara la interacción) y la revolución que supuso el auge de los blogs, wikis, foros, las redes sociales y otras herramientas relacionadas.

Todos estos sitios utilizan la inteligencia colectiva para proporcionar servicios interactivos en la red donde el usuario tiene control para publicar sus datos y compartirlos con los demás; cuyos contenidos son un tanto subjetivo, cargado de opiniones y valoraciones, que puede ser de gran utilidad para recomendación de una marca o producto determinado.

Solo restaba entonces, entender el valor de la información generada por estos medios y a través de este *feedback*[6] ajustar estrategias comerciales, mejorar los servicios, detectar tendencias para lanzar nuevos productos y recomendar otros. Eso, ligado al desafío de automatizar los procesos de recolección de información y filtrado de ruido con objeto de encontrar información relevante dentro a la multitud de datos.

La nueva forma de hacer Internet, conocida como Web Social [16], ha despertado el interés de aquellos, que están enfocando sus esfuerzos en comprender los métodos para determinar los sentimientos en el contexto de las comunicaciones sociales.

[6] *Retroalimentación. Respuesta u opinión sobre un asunto determinado.*

El análisis de sentimiento a través de las redes sociales ayuda a entender el comportamiento y gustos de los usuarios y puede ser utilizado para sondear la opinión pública general sobre ciertos temas, como componente en sistemas de recomendación y también como medio de inteligencia de negocio con el fin de prevenir abandono de cliente, compararse con la competencia o anticipar eventos futuros.

Determinar y clasificar sentimientos de un fragmento de texto como una publicación de una red social es más complicado aún; ya que muchas veces depende del contexto en el que se lee el texto, puede incurrir también las variaciones culturales, jergas, faltas de ortografía; u otros factores como la dificultad de identificar ironías.

Redes sociales

Las **redes sociales** son sitios de internet que permiten a las personas conectarse con sus amigos e incluso realizar nuevas amistades, de manera virtual, y compartir contenidos, interactuar, crear **comunidades** sobre intereses similares: trabajo, lecturas, juegos, amistad, relaciones amorosas, relaciones comerciales, etc.

Es una estructura capaz de comunicar entre sí a personas o instituciones, sin necesariamente tener que conocerse de antemano para relacionarse; sino que pueden hacerlo a través de la red social.

Las redes sociales, nacen a raíz de la puesta en marcha de los *Networking Services* (SNS)[7], y tienen su origen en el año **1995** cuando el estadounidense Randy Conrads creó el sitio web *Classmates*[8], para ese entonces y en la actualidad pretendía que la gente pudiera recuperar o mantener el contacto con antiguos compañeros del colegio, instituto, universidad, trabajo, entre otros.

Dos años después, nace la red *SixDegrees*, que permitía crear un perfil social y una lista de amigos, características muy similares al concepto que tenemos actualmente de red social. Fue llamada así por la teoría de los seis grados de separación, que se basaba en un cuento llamado Chains [15], publicado en 1929. El término fue popularizado por el sociólogo *Duncan Watts*. Según esta teoría una persona, independientemente del lugar del planeta en el que viva, está conectada a otra, a través de una **red de conocidos que no superan los 5 intermediarios**.

Las SNS pueden clasificarse según su origen y función en profesionales, temáticas y las genéricas; estas últimas son las más numerosa y populares, entre las que se destacan están Facebook[9], Twitter[10] y Google+[11].

El éxito de las redes sociales ha sido imparable, para ser protagonistas de este fenómeno social que ha cambiado y revolucionado la forma de comunicación e interacción con los seres humanos, se hace necesario reinventarnos, para de esta forma

[7] *Plataformas para construir redes sociales o las relaciones sociales entre personas que comparten intereses, actividades o ideas.*

[8] *http://www.classmates.com/*

[9] *https://www.facebook.com*

[10] *https://twitter.com*

[11] *https://plus.google.com*

sacarle el mejor partido a dichas herramientas.

Twitter

Twitter es un servicio de comunicación bidireccional con el que puedes compartir información de diverso tipo de una forma rápida, sencilla y gratuita. Es una de las redes de *microblogging* más populares que existen en la actualidad y su éxito reside en el envío de mensajes cortos llamados *tweets*.

Fue creada por Jack Dorsey y su equipo en **2006** y la idea se inspira en el envío de fragmentos cortos de texto de 140 caracteres ***¿Qué está pasando?*** es la pregunta de esta red social, que en apenas unos años pasó a ser uno de los servicios de redes sociales más elegidos.

En Twitter te llega **mucha información de calidad en poco tiempo**. Hace además un papel muy bueno, el de guía, que con la ayuda de las opiniones y recomendaciones de los usuarios te lleva rápidamente **a las mejores fuentes de información** sobre prácticamente cualquier tema.

Por otra parte, el formato tipo *Short Message Service* (SMS), lejos de ser un inconveniente, ha resultado ser un factor muy positivo porque ha sido precisamente la razón fundamental que ha permitido esa agilidad y eficiencia tan característica. Se puede decir que Twitter ha creado una cultura de la "eficiencia" entre sus usuarios, una cultura de aprovechar al máximo el espacio disponible e ir al grano, de distinguir el grano de la paja, lo cual es, sin duda, un gran valor añadido.

Dada la naturaleza de los mensajes breves, comenzaron a adoptarse una serie de convenciones permitiendo ahorrar tanto en tiempo como espacio (caracteres), haciendo un uso más eficaz de herramienta. Por tratarse sobre todo del empleo de un lenguaje escrito con rasgos de la inmediatez [17] y espontaneidad, el cual se provee escasa planificación, incluye el uso de abreviaturas y acrónimos de palabras, como "RT" para *"ReTweet"*; menciones, que permite indicar el destinatario para un mensaje determinado, con el uso del signo "@", seguido de un nombre de usuario y emoticonos[12], que muestran el sentimiento del usuario.

En un *tweet*, se pueden incluir además enlaces, que permiten acceder a sitios web, vídeos e imágenes recomendados por el usuario, con frecuencia se hace uso de etiquetas o *hashtags*, simbolizado con el signo "#", seguido de una palabra o un grupo de palabras (sin espacio entre ellas), para enriquecer el mensaje y de esta forma marcar el tema del *tweet* y servir de metadatos para las búsquedas o ser indicado como un tema global (*trending topic*), cuando aparece en un número elevado de *tweet*.

A diferencia de otras redes sociales, es clasificada como una plataforma de comunicación asimétrica, por lo cual no se requiere un consentimiento mutuo entre los implicados ni se necesita que exista una relación recíproca entre los usuarios para que se conecten y compartan información. Debido a ello se definen los términos de

[12] *Neologismo que proviene de emoción e icono.*

following o *seguidor* y *follower* o *seguido*. El primero de ellos se le atribuye al grupo de personas a que un usuario determinado sigue. A su vez, estos usuarios seguidos pasan a tener un nuevo seguidor, un *follower*. Cuando dos usuarios se siguen mutuamente, se les puede considerar *"amigos"*.

Por otra parte, en el *timeline*[13] o línea de tiempo, se pueden visualizar los mensajes de interés para usuario, evitando, de esa forma, la difusión de contenidos no deseados.

Twitter ofrece diversas aplicaciones que permiten desde buscar noticias o eventos hasta encontrar trabajo, pero también existen infinidad de aplicaciones online que amplían sus posibilidades y que van más allá de toda expectativa, como la *Application Programming Interface* (API), contribuyendo así a que se desarrollen aplicaciones de terceros, que se conectan e interactúan con la red social, posibilitando así poder extraer información en tiempo real, analizarlas para entender patrones de consumo y tendencias de comportamiento de los usuarios.

Inteligencia Artificial

La Inteligencia Artificial (IA) es una disciplina académica relacionada con la teoría de la computación cuyo objetivo es emular algunas de las facultades intelectuales humanas en sistemas artificiales [18]. El concepto de IA existe desde mediados del siglo pasado, en la década del **1960**, y engloba todas las metodologías y tecnologías para abordar el desafío de **"dotar a una máquina de inteligencia"**. Dando solución así a problemas de carácter abstractos como la demostración de teoremas matemáticos, la adquisición del lenguaje, el jugar al ajedrez o la traducción automática de textos.

El diseño de un sistema de IA normalmente requiere la utilización de herramientas de disciplinas muy diferentes como el cálculo numérico, la estadística, la informática, el control automático, la robótica o la neurociencia. En algunos aspectos, además, se beneficia de investigaciones en áreas tan diversas como la psicología, la sociología o la filosofía [18].

En ocasiones, los sistemas de IA resuelven problemas de forma *heurística* mediante un procedimiento de ensayo y error que incorpora información relevante basada en conocimientos previos. Cuando un mismo problema puede resolverse mediante sistemas naturales (cerebro) o artificiales (computadora), los algoritmos que siguen cada implementación suelen ser completamente diferentes puesto que el conjunto de instrucciones elementales de cada sistema son también diferentes. El cerebro procesa la información mediante la activación coordinada de redes de neuronas en áreas especializadas (cortex visual, cortex motor, etc.). En el sistema nervioso, los datos se transmiten y reciben codificados en variables como la frecuencia de activación de las neuronas o los intervalos en los que se generan los potenciales de acción neuronales. El elevado número de neuronas que intervienen en un proceso de computación natural hace que las fluctuaciones fisiológicas tengan un papel relevante y que los procesos

[13] *Muro cronológico de mensajes publicados por el usuario y por las personas que éste sigue.*

computacionales se realicen de forma estadística mediante la actividad promediada en subconjuntos de neuronas [18].

En un sistema IA, en cambio, las instrucciones básicas son las propias de una computadora, es decir operaciones aritmetico-lógicas, de lectura/escritura de registros y de control de flujo secuencial [18].

Algunas de las principales diferencias entre los sistemas de inteligencia artificial y natural se muestran en la tabla 1:

Nivel	**Natural**	**Artificial**
Abstracción	Representación y manipulación de objetos abstractos	Representación y manipulación de objetos abstractos
Computacional	Activación coordinada de áreas cerebrales	Algoritmo/procedimiento efectivo
Programación	Conexiones sinápticas plasticidad	Secuencia de operaciones artimético-lógicas
Arquitectura	Redes excitatorias e inhibitorias	CPU + memoria
Hardware	Neurona	Transistor

Tabla 1: Comparación entre inteligencia natural y artificial

Podría decirse que a pesar de las enormes diferencias entre *sistemas naturales* y *sistemas artificiales*, a un cierto nivel de abstracción ambos pueden describirse como **sistemas de procesado de objetos abstractos** mediante un conjunto de reglas.

La universalidad de las máquinas dió un empuje a la IA, generando diferentes ramas de desarrollo; por una parte se encuentra, la ***IA convencional*** conocida como ***IA simbólico-deductiva*** e ***IA débil***; agrupa *razonamiento basado en casos*, *sistemas expertos*, *árboles de decisión*, *redes bayesianas* e *inteligencia artificial basada en comportamientos*, proporcionándole a los sistemas complejos, autonomía pudiendo así auto-regularse y controlarse.

Por su parte la ***IA computacional***, también conocida como ***IA subsimbólica-inductiva*** e ***IA fuerte***, implica desarrollo o aprendizaje interativo[14], el mismo se realiza basándose en datos empíricos. Algunos métodos de esta rama incluyen *máquina de vectores soporte (sistemas que permiten reconocimiento de patrones genéricos de gran potencias)*, *modelos ocultos de Markov (aprendizaje basado en dependencia temporal de eventos probabilísticos)* y *redes neuronales* (ANN).

La *IA computacional*, es la que se centra en el desarrollo de sistemas capaces de adaptarse. Genéricamente hablando, se introduce el concepto de ***aprender*** de manera dinámica para adaptar el comportamiento.

[14] *Enfoque pedagógico que incluye el uso de sistemas tecnológicos. Complementa cualquier área curricular.*

Los sistemas de aprendizaje ***Machine Learning (ML)/ Deep Learning (DL)*** están fuertemente basados en ANN y se alimentan de gran cantidad de datos *(Big-Data)* para generar la capacidad de adaptar el comportamiento.

Aunque los términos se utilizan a veces como sinónimos, el DL y el ML no son lo mismo, siendo el primero un tipo particular del segundo, es decir, el DL es ML, pero existen técnicas de ML que no son DL [19].

Machine Learning: Autoaprendizaje

El ML, se diferencia de conceptos tradicionales por la capacidad que tiene de otorgar a los algoritmos el poder aprender de los datos. En este punto, el ML forma parte de la inteligencia artificial, pues su desarrollo se basa en la teoría del aprendizaje computacional. [19].

Pero también bebe de otras fuentes, ya que se desarrolló a partir del estudio de **reconocimiento de patrones**[15], una ciencia que se ocupa de los procesos sobre ingeniería, computación y matemáticas relacionados con objetos. Su objetivo es obtener información que permita establecer propiedades de entre los conjuntos de dichos objetos.

El *aprendizaje* se divide principalmente en dos tipos: **Aprendizaje con Profesor** o **Supervisado** y **sin Profesor** o **No Supervisado** [20].

Los sistemas de ***clasificación supervisados*** son aquellos en los que, a partir de un conjunto de ejemplos clasificados *(conjunto de entrenamiento)*, intentamos asignar una clasificación a un segundo conjunto de ejemplos. Por otra parte, los sistemas de ***clasificación no supervisados*** son aquellos en los que no disponemos de una batería de ejemplos previamente clasificados, sino que únicamente a partir de las propiedades de los ejemplos intentamos dar una agrupación *(clasificación, clustering)* de los ejemplos según su similaridad.

En el presente trabajo se optó por utilizar la variante del *aprendizaje no supervisado*, así como algunos algoritmos del mismo. En concreto nos centraremos en los métodos basados en distancia (Clustering), aunque puede que se apliquen otros métodos, ya sean los basados en modelos probabilísticos (Naïve Bayes) para eliminar datos irrelevantes, métodos basados en reglas (Árboles de Decisión) para poder realizar preselecciones de manera más objetiva o métodos basados en *kernels* (Máquina de Vectores de Soporte) para trabajar con textos aleatorios y poder realizar categorizaciones.

Naïve Bayes

Naïves bayes (NB) está categorizado entre los *métodos basados en modelos probabilísticos* ya que suele estimar un conjunto de parámetros que expresan la probabilidad condicionada de cada clase, dadas las propiedades de un ejemplo (descrito

[15] *http://cgm.cs.mcgill.ca/ godfried/teaching/pr-web.html.*

en forma de atributos). A partir de entonces, estos parámetros pueden ser combinados para asignar las clases que maximizan sus probabilidades a nuevos ejemplos [18].

El NB, está basado en el teorema de *Bayes*, es considerado el representante más simple de los algoritmos basados en probabilidades.

El algoritmo de NB clasifica nuevos ejemplos $x = (x_1, ..., x_m)$ asignándole la clase k que maximiza la probabilidad condicional de la clase dada la secuencia observada de atributos en la ecuación 1.

$$arg_k max P(k|x_1, ..., x_m) = arg_k max \frac{P(x_1, ..., x_m|k)P(k)}{P(x_1, ..., x_m)} \approx arg_k max P(k) \prod_{t=1}^{m} P(x_t|k) \tag{1}$$

donde $P(k)$ y $P(x_t|k)$ se estiman a partir del conjunto de entrenamiento, utilizando las frecuencias relativas (estimación de la máxima verosimilitud).

Durante el proceso de entrenamiento se comienza calculando $P(k)$ para cada una de las clases $k \epsilon Y$. Se aplica una estimación de la máxima verosimilitud. Posteriormente se calcula $P(x_t|k)$ para cada pareja *atributo-valor* y para cada clase, terminando así el proceso de entrenamiento. A partir de aquí, si llega un ejemplo nuevo se tendrá que aplicar la fórmula $arg_k max P(k) \prod_{t=1}^{m} P(x_t|k)$ para clasificarlo.

Uno de los problemas de NB, es que el algoritmo asume la independencia de los diferentes atributos que representan a un ejemplo [18].

En el caso de tener un problema representado con algún atributo continuo, como los numéricos, es necesario algún tipo de proceso, como por ejemplo aplicando una categorización, dividiendo el continuo en intervalos o asumiendo que los valores de cada clase siguen una *distribución gaussiana* [18] y aplicar la ecuación 2:

$$P(x = v|c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} exp\left(-\frac{(v - \mu_c)^2}{2\sigma_c^2}\right) \tag{2}$$

donde x corresponde al atributo, v a su valor, c a la clase, μ_c al promedio de valores de la clase c y σ_c^2 a su desviación estándar.

En caso de que se obtenga un ejemplo de *test*, con alguna pareja *atributo-valor* que no ha aparecido en el conjunto de entrenamiento, y no se tenga ningún valor para $P(x_t|k)$ se pueden emplear **técnicas de suavizado**.

Un ejemplo de *técnica de suavizado* para este algoritmo consiste en sustituir la $P(x_t|k)$ que contiene el contador a cero por $P(k)/n$ donde n corresponde al número de ejemplos de entrenamiento.

Otra característica a destacar es que cuando el conjunto de entrenamiento no está balanceado, tiende a clasificar los ejemplos hacia la clase que tiene más ejemplos dentro del conjunto de entrenamiento [18].

Clustering dentro de clases

Clustering se traduce como *agrupamiento o clasificación*. El objetivo del clustering no es clasificar, estimar o predecir una variable; sino entender la estructura macroscópica y relaciones entre objetos, considerando las maneras en las que estos son similares y diferentes. En otras palabras, se enfoca en segmentar el conjunto completo de datos en subgrupos homogéneos; y a los objetos con cierta similaridad se los agrupa en cluster.

El algoritmo de agrupamiento *Clustering (k-means)* busca una partición de los datos tal que cada punto esté asignado al grupo cuyo centro (llamado centroide, pues no tiene por qué ser un punto de los datos) sea más cercano. Se le debe indicar el número k de clusters deseado, pues por sí mismo no es capaz de determinarlo [18].

Pero para determinar a qué grupo pertenece cada instancia de datos, se hace necesario medir las distancias entre cada instancia. De ahí que se utilice ***Clustering* dentro de clases** para cacular *la fórmula de la distancia* que más nos acerque a los resultados esperados.

El ***Clustering* dentro de clases** consiste en aplicar un algoritmo de categorización para calcular cierto número de centroides para cada una de las clases que aparece en el conjunto de entrenamiento. Una vez hecho esto, utiliza el **kNN** (k vecinos más cercanos)[16] seleccionando todos los centroides como conjunto de entrenamiento y aplica el *1NN* para la fase de clasificación.

Para obtener el conjunto de los k vecinos más cercanos, se calcula la distancia entre el ejemplo a clasificar $x = (x_1, ..., x_m)$ y todos los ejemplos guardados $x_i = (x_{i1}, ..., x_{im})$ [18]. La distancia más utilizada es la euclídea, y se define tal cual la ecuación 3:

$$de(x, x_i) = \sqrt{\sum_{j=1}^{m}(x_j - x_{ij})^2} \tag{3}$$

El proceso de entrenamiento consiste en guardar los datos y en caso de clasificar un ejemplo nuevo, tenemos que calcular las distancias entre el nuevo ejemplo y todos los del conjunto de entrenamiento.

Un aspecto a tener en cuenta tiene que ver con la eficiencia computacional; ya que el algoritmo realiza todos los cálculos en el proceso de clasificación. Así, aun siendo un método rápido globalmente, tenemos que tener en cuenta que el proceso de clasificación no lo es. Esto puede llegar a ser crítico para aplicaciones de tiempo real que necesiten una respuesta rápida [18].

[16] *En inglés, k nearest neighbours.*

Árboles de decisión

Los **Árboles de decisión** forman parte de los *métodos basados en reglas*. Dichos métodos adquieren reglas de selección asociadas a cada una de las clases. Dado un ejemplo de *test*, el sistema selecciona la clase que verifica algunas de las reglas que determinan una de las clases.

Un árbol de decisión es una forma de representar reglas de clasificación inherentes a los datos, con una estructura en árbol n-ario que particiona los datos de manera recursiva. Cada rama de un árbol de decisión representa una regla que decide entre una conjunción de valores de un atributo básico *(nodos internos)* o realiza una predicción de la clase *(nodos terminales)* [18].

El algoritmo básico está pensado para trabajar con atributos nominales. El conjunto de entrenamiento queda definido por $S = (x_1, y_1), ..., (x_n, y_n)$, donde cada componente x corresponde a $x_t = (x_{t1}, ..., x_{tm})$ donde m corresponde al número de atributos de los ejemplos de entrenamiento; y el conjunto de atributos por $A = \{a_1, ..., a_m\}$ donde dom(aj) corresponde al conjunto de todos los posibles valores del atributo aj, para cualquier valor de un ejemplo de entrenamiento $x_{tj}\epsilon$ dom(a_j) [18].

El proceso de construcción del árbol es iterativo; en cada iteración, se selecciona el atributo que mejor particiona el conjunto de entrenamiento. Para realizar este proceso, se tiene en cuenta la *bondad de las particiones* que genera cada uno de los atributos y en un segundo paso, seleccionar el mejor. La partición del atributo a_j genera $|\text{dom}(a_j)|$ conjuntos, que corresponde al número de elementos del conjunto.

Una medida para mirar la bondad de una partición, consiste en asignar a cada conjunto de la partición la clase mayoritaria del mismo y contar cuántos quedan bien clasificados y dividirlo por el número de ejemplos. Una vez calculadas las bondades de todos los atributos, escogemos el mejor. Cada conjunto de la mejor partición pasará a ser un nuevo nodo del árbol. A este nodo se llegará a través de una regla del tipo *atributo = valor*. Si todos los ejemplos del conjunto han quedado bien clasificados, lo convertimos en *nodo terminal* con la clase de los ejemplos. En caso contrario, lo convertimos en *nodo interno* y aplicamos una nueva iteración al conjunto ("reducido") eliminando el atributo que ha generado la partición. En caso de no quedar atributos, lo convertiríamos en *nodo terminal* asignando la clase mayoritaria. Para realizar el *test*, exploramos el árbol en función de los valores de los atributos del ejemplo de *test* y las reglas del árbol hasta llegar al *nodo terminal*, y damos como predicción la clase del *nodo terminal* al que lleguemos [18].

Este algoritmo tiene la gran ventaja de la facilidad de interpretación del modelo de aprendizaje, así mismo, tiene varios inconvenientes, como la elevada fragmentación de los datos en presencia de atributos con muchos valores y el elevado coste computacional que esto implica. Por lo cual no es muy adecuado para problemas con grandes espacios de atributos; como pueden ser aquellos que contienen información léxica, la categorización de textos o los filtros anti spam. Además de que los nodos terminales correspondientes a reglas que dan cobertura a pocos ejemplos de entrenamiento no producen estimaciones fiables de las clases y tiende a sobreentrenar el

conjunto de entrenamiento[17] [18]. Para suavizar este efecto se puede utilizar alguna técnica de poda[18].

Dicho algoritmo fue diseñado para tratar con atributos nominales, pero existen alternativas para el tratamiento de atributos numéricos. El más común se basa en la utilización de puntos de corte. Estos son un punto que divide el conjunto de valores de un atributo en dos (los menores y los mayores del punto de corte) [18].

El mejor punto de corte de un atributo numérico se obtiene, ordenando los valores y eliminando los elementos repetidos. Luego se calculan los posibles puntos de corte como el promedio de cada dos valores consecutivos y finalmente se calcula el mejor de ellos como aquel con mejor bondad tenga.

Máquinas de vectores de soporte

Las **Máquinas de Vectores de Soporte (SVMs)**[19] están clasificados dentro de *Clasificadores lineales y métodos basados en kernels*, que mejoran el clasificador lineal buscando un mejor hiperplano que el que se genera con el clasificador lineal.

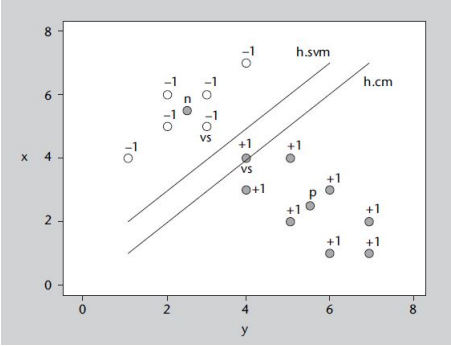


Figura 1: Hiperplanos lineales y SVM: tomada de [18]

Las SVMs son algoritmos de optimización que escoge el hiperplano con margen máximo de entre todos los posibles hiperplanos que separan los ejemplos positivos de los negativos *(el que tiene la misma distancia a los ejemplos positivos que a los negativos)*. Describe el hiperplano a partir de los llamados vectores de soporte. Estos suelen ser los puntos más cercanos al hiperplano y los que lo definen. En la fig. 1, *h.svm* corresponde al hiperplano de margen máximo que encontrarían las SVM para este conjunto y los vectores de soporte están marcados con *vs*.

Dicho algoritmo es un método de aprendizaje que se basa en la *maximización del margen* (Ver fig. 2), tomando como estrategia la búsqueda los vectores de soporte. Este mecanismo de búsqueda parte de un espacio de hipótesis de funciones lineales en un espacio de atributos altamente multidimensional. En él se entrena con

[17] *Este efecto se conoce en inglés como overfitting.*

[18] *En inglés, prunning.*

[19] *Support Vector Machines: N. Cristianini; J. Shawe-Taylor (2000). An Introduction to Support Vector Machines (SVMs). Cambridge University Press.*

un algoritmo de la teoría de la optimización derivado de la teoría del aprendizaje estadístico [18].

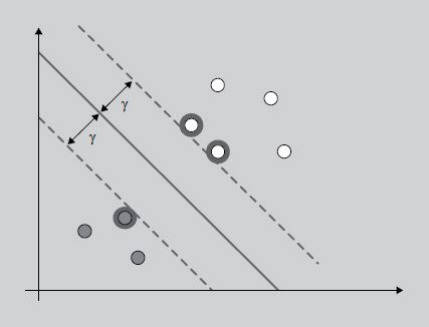


Figura 2: Maximización del margen: tomada de [18]

Entrenar una SVM con un conjunto mínimo de entrenamiento es un reto. Esto se debe tener en cuenta porque cuanto mayor es el conjunto, mayor es el coste del aprendizaje. Alcanzar altos niveles de fiabilidad/coste debe ser el propósito a obtener.

Redes Neuronales Artificiales

Las redes neuronales no son una idea nueva. Datan de los años **40** y **50**, cuando se empezaron a publicar los primeros conceptos. Sin embargo, nunca tuvieron un gran éxito, debido a que se necesita una cantidad importante de recursos de un ordenador o una gran cantidad de datos para entrenar y ejecutar una red neuronal con buenos resultados.

El concepto de **Red Neuronal Artificial (RNA)**, es un modelo matemático inspirado en el comportamiento biológico de las neuronas y en cómo se organizan formando la estructura del cerebro [20]. Consisten en un gran número de elementos simples de procesamiento llamados nodos o neuronas que están organizados en capas [21].

Una **neurona artificial** pretende mimetizar las características más importantes de la neurona biólogica. En general, recibe las señales de entrada de las neuronas vecinas ponderadas por los pesos de las conexiones. La suma de estas señales ponderadas proporciona la entrada total o neta de la neurona y, mediante la aplicación de una función matemática denominada función de salida, sobre la entrada neta, se calcula un valor de salida, el cual es enviado a otras neuronas (Ver fig. 3). Tanto los valores de entrada a la neurona como su salida pueden ser señales excitatorias (cuando el valor es positivo) o inhibitorias (cuando el valor es negativo) [21].

Cada neurona está conectada con otras neuronas mediante enlaces de comunicación, cada uno de los cuales tiene asociado un peso. Los pesos representan la información que será usada por la red neuronal para resolver un problema determinado [21].

El objetivo es encontrar la combinación que mejor se ajusta entrenando a la red

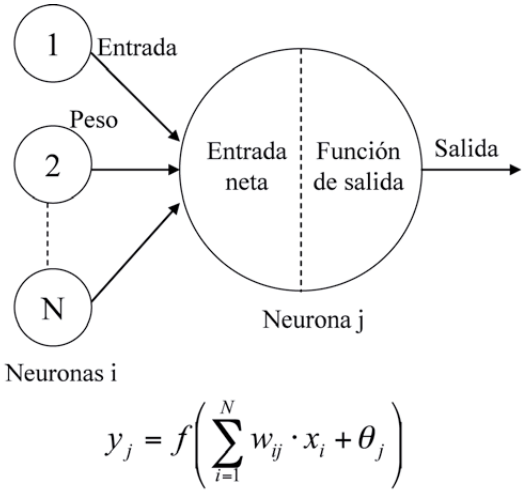


Figura 3: Funcionamiento general de una neurona artificial: tomada de [46]

neuronal. Este entrenamiento, aprendizaje, es la parte crucial, ya que nos marcará la precisión del algoritmo. Mediante este entrenamiento o aprendizaje, las **RNA** crean su propia representación interna del problema, por tal motivo se dice que son *autoorganizadas*. Posteriormente, pueden responder adecuadamente cuando se les presentan situaciones a las que no habían sido expuestas anteriormente, generalizando casos anteriores a casos nuevos. Esta característica es fundamental ya que permite a la red responder correctamente no sólo ante informaciones novedosas, sino también ante informaciones distorsionadas o incompletas [21].

En este sentido se usarán los *Self Organizing Maps*[20] (SOM), que son un tipo de RNA que se entrena utilizando un aprendizaje no supervisado para producir una representación discretizada de baja dimensión (generalmente bidimensional) del espacio de entrada de las muestras de entrenamiento; reduciendo la dimensionalidad.

Los SOM difieren de otras redes neuronales artificiales, ya que aplican el aprendizaje competitivo en lugar del aprendizaje con corrección de errores (como las Redes de Retropropagación Supervisadas). Además usan una función de vecindad para preservar las propiedades topológicas del espacio de entrada [44].

Una red ya entrenada con el SOM se puede usar luego para hacer predicciones o clasificaciones, pudiéndose emplear para clasificar el sentimiento y posteriormente representar los resultados. Una ventaja del SOM sobre otros métodos es que el propio paso de clasificación produce un mapa gráfico donde las clases relacionadas aparecen cerca unas de otras, por lo que se puede transmitir más información.

Trabajos relacionados

Los ejemplos en el campo de IA van más allá de la robótica, abarcando desde *aplicaciones en data science* dentro del *Big Data*, hasta, programas informáticos de

[20] *En español, Mapas Auto-Organizados*

detección de fraude, aplicación en el sector de la medicina y en numerosos ámbitos de la investigación, pero en el campo de **Recursos Humanos (RRHH)** y **Selección de Personal** se ha hecho muy poco. Podría decirse que no existen trabajos previos de análisis de sentimientos en el campo de IA; sólo las grandes tecnológicas están en ello, pero sin publicar abiertamente nada, como es el caso de Google[21].

Para finales de **junio del 2017**, **Google** lanzó una nueva función de búsqueda de empleo en sus páginas de resultados de búsqueda que le permite buscar empleos en prácticamente todas las principales bolsas de trabajo en línea como LinkedIn, Monster, WayUp, DirectEmployers, CareerBuilder y Facebook y otros. Google también incluirá listados de trabajo que encuentre en la página de inicio de una compañía [22]. Para crear esta lista completa, primero se tienen que eliminar todas las listas duplicadas que los empleadores publican en todos estos sitios de trabajo. Luego, sus *algoritmos entrenados en el aprendizaje automático* los examinan, para luego categorizarlos.

Con esta nueva funcionalidad añadida al buscador, Google no intenta filtrar los trabajos según lo que ya sabe; el hecho de que a una persona le guste ir de pesca no significa que esté buscando trabajo en un barco de pesca [22].

Sushant Shankar y **Irving Lin** [23] por su parte crearon un clasificador de productos mediante el uso de *aprendizaje automático supervisado*. Dado un producto cualquiera y en dependencia de la descripción, modelo, SKU[22], entre otras características; lo agrupan en una categoría particular con productos similares. Para ello, analizan la información relacionada del catálogo de productos de los diferentes distribuidores en Amazon[23] y en función de eso crean un clasificador.

Teniendo en cuenta que las fuente de datos predominante era en forma de texto, primeramente se aplicaron algunas técnicas estándar en el procesamiento de texto como *stemming/ lemmatization*, *lowecasing*, *eliminar stopwords* y se implementó el clasificador de NB, obteniéndose una precisión de aproximadamente el 75 %. Posteriormente tras incluir múltiples clasificaciones aumentó entonces la precisión de alrededor del 91 %.

Aunque el trabajo anterior está pensado para clasificar productos (físicos), podría decirse que no se aleja tanto de nuestra propuesta (clasificar candidatos a un puesto de trabajo), ya que se capturan datos desde una web, realizan procesamiento de datos y posteriormente realizan una clasificación haciendo uso de aprendizaje automático.

En [26] se proponen algunos mecanismos de aprendizaje automático como técnicas de NLP; clasificador de NB, Árboles de decisión, SVM determinan clasificadores de texto; centrándose principalmente en la reducción del cálculo computacional y en el costo que ello implica.

De igual forma trabajó la categorización de textos Thorsten Joachims [27] esta vez comparando el rendimiento de las *SVM*, usando núcleos polinomiales con algunos métodos de aprendizaje convencionales como *Naïve Bayes*, *Árboles de decisión* y

[21] *https://www.google.es/*
[22] *Stock-keeping unit - Código de artículo o número de referencia*
[23] *https://www.amazon.com*

otros como el *kNN*. Después de la implentación se pudo apreciar que en comparación con los métodos convencionales, todas las SVM funcionan mejor independientemente de la elección de los parámetros. Incluso para espacios de hipótesis complejas, como los polinomios de grado 5, no se produce sobreposición a pesar de utilizar todas las características.

Los resultados muestran que esta estrategia es adecuada para elegir una buena configuración automática de parámetros, ya que alcanzó un valor de confianza sobre el 86.0 % para el SVM polinomial, superando así al resto de los métodos utilizados. Además de que las SVM son más costosas que NB y kNN; y más rápidas que KNN en el momento de la clasificación.

Tripathy Abinash [28] por su parte propone una clasificación de texto para determinar la intención del autor del texto, a partir del *análisis de sentimientos*. Teniendo en cuenta que la intención puede ser de tipo admiración (positiva) o crítica (negativa), presenta una comparación de los resultados obtenidos al aplicar el algoritmo de clasificación NB y SVM. El conjunto de datos considerado para el entrenamiento y pruebas del modelo fueron etiquetados según su polaridad y se realizó una comparación con los resultados disponibles en la literatura existente para un examen crítico.

Los resultados obtenidos demuestran que entre el conjunto de autores en su mayoría, los valores de las SVM superan a NB; obteniéndose valores, en el caso del más bajo *(Pang, Lee, y Vaithyanathan)* [14] con un 82.9 % y el valor más alto *(Zhang, Xu, Su, y Xu)* [13] con un 90.30 %.

Dada las características de las SVM, en la mayoría de los procesos de clasificación se observa que produce un resultado comparativamente convincente [28].

El presente trabajo se enfocará en un análisis a nivel de sentencia haciendo uso de algunas métodos de NLP para clasificación por polaridad, y se utilizarán algunas de las técnicas de aprendizaje automático antes mencionadas, con el objetivo de clasificar textos cortos (*tweets*) y enfocar esfuerzos en mejorar el poder de predicción de sentimiento de esos textos. Posteriormente, cada candidato a un puesto de trabajo será clasificado en función de ese sentimiento, siendo este último un elemento a tomar en cuenta pero no determinante.

Metodología

En esta sección se describe el análisis del problema, las herramientas utilizadas, así como también, se detalla la implementación del sistema.

Análisis del Problema

El Grupo Faster[24] es una empresa con casi 30 años de experiencia en el sector de RR.HH. y con más de 30 delegaciones en toda España; Faster trabaja para las empresas que buscan empleados, ofreciéndoles de una forma eficiente y ágil cubrir sus necesidades de personal. Y para personas que desean mejorar su empleabilidad, facilitándoles el acceso a un empleo con todas las garantías legales.

Está formado además por cuatro compañías, entre la que se encuentra Faster Empleo ETT SA[25]. Dicha compañia está dedicada al trabajo temporal y es a su vez la empresa más representativa del grupo.

A través de Faster Empleo ETT SA se realiza toda la labor de RR.HH. y se manejan a su vez varios perfiles en las redes sociales como Facebook, Linkedin y Twitter que hacen de la selección de personal y la gestión comercial una de las labores más importantes.

El presente trabajo surge de la necesidad, de conocer el estado de ánimo de los seguidores de *@FasterEmpleo* en Twitter, se asume entonces que dichos seguidores, pueden ser de dos tipos: clientes o empleados, y que nos siguen en dicha red para estar al corriente de las ofertas laborales que allí se publican.

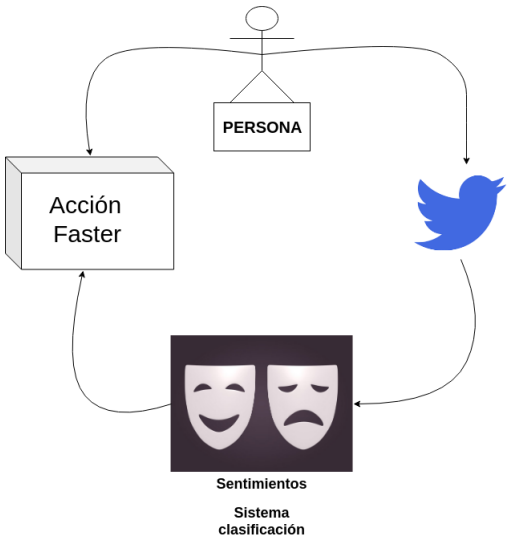


Figura 4: Sentimientos, sistema de clasificación y acciones comerciales

[24] *https://faster.es/*
[25] *https://ats.bizneo.com/trabajar/faster*

Una vez se tenga la clasificación de los mensajes de Twitter, solo restaría relacionar dichos perfiles con los usuarios del sistema; para en base a ese resultado tomar acciones comerciales que ayuden a la mejora continua del Grupo (Ver Figura 4); de ahí que nos diéramos a la tarea de crear una herramienta que analice dichos sentimientos.

Una herramienta de análisis de sentimiento debe tener la capacidad de procesar un texto y clasificarlo independientemente de la polaridad de sentimientos que expresa su contenido.

Haciendo uso de varias técnicas de IA, este trabajo tiene como principal objetivo crear un modelo de clasificación del "estado de ánimo" de un grupo cerrado de usuarios para la toma de decisiones.

Para llegar a la clasificación y para este trabajo en específico, se realizaron las etapas que se muestran en la Figura 5:

Figura 5: Etapas empleadas en la herramienta de análisis de sentimiento

Extracción de datos

Uno de los requisitos del cliente final es poder almacenar toda la información relacionada con los mensajes (*tweets*) de los seguidores de la cuenta de Faster en Twitter: *@FasterEmpleo*, para correlacionarla con las bases de datos del Grupo Faster, y de esta forma se nutran tanto la herramienta de análisis de sentimiento, como la propia base datos.

Para ello, haciendo uso de la API de Twitter y a través de Tweepy[26], accedemos al *timeline* y recopilamos todos los *tweets* en crudo, sin aplicar ningún tipo de filtrado o procesado.

Una vez recopilada la información; tomamos los datos que desde nuestro punto de vista, eran relevantes o aportaban algún valor al desarrollo en cuestión, como: nombre de usuario, *tweet*, localización, número de seguidores, fecha publicación, fuente, entre otras. Sin descastar que para futuros desarrollos se añadan otros campos a nuestro conjunto de datos.

Posteriormente, se definieron los modelos en la base datos en los cuales se iban a almacenar los datos recopilados de Twitter, algunos de los más significativos son:

- **Modelo para *tweets*** (Ver Figura 6): Contiene todos los datos relacionados con el tweet; como texto (*tweet*), tipo de tweet (type), fecha creación

[26] *https://www.tweepy.org*

(date_create), fuente (source), localización (location), y una *foreign key* de usuario, de forma tal que queden relacionados entre sí.

- **Modelo para usuarios:** Contiene toda la información relacionada con un usuario; como id usuario (id_user), nombre usuario (scree_name), foto, descripción (description), sentimiento (feeling), etc.

- **Modelo para Persona (Persona Física o Jurídica):** Contiene toda la información relacionada con una persona; como nombre (screen_name), localización (location), seguidores (follows), tipo persona (type_person), y además una *foreign key* de usuario, para así relacionar ambos modelos.

Se adicionaron otros campos al modelo para los *tweets*; dichos valores irán de la mano del sentimiento de los *tweets*, y nos indicarán datos del tipo contextuales, como el año de publicación del mismo, localidad, horario en que fue publicado (mañana, tarde, noche), tipo de persona (Persona Física, Persona Jurídica), tipo de publicación (*tweet*, *retweet*), entre otras.

tweet	type	date_create	date_published	source	source_url	location
RT @largocaballerof: Exposición "130 años de	Retweet	2009-10-30 19:26:44	2019-05-13 08:26:00	Twitter Web App	https://mobile.twitter.com	España
Talent #Engagement: ¿Cómo promoverlo en	Tweet	2011-03-16 11:53:43	2019-05-13 09:00:00	HubSpot	http://www.hubspot.com/	Salamanca
RT @instrabajoyss: 1 ¿Qué es el registro obli	Retweet	2009-08-16 18:51:40	2019-05-13 08:59:57	Twitter for iPhor	http://twitter.com/downlc	París, Francia
Los sindicatos de Endesa plantean otra huelg	Tweet	2009-10-30 19:26:44	2019-05-13 08:59:45	Twitter Web Clie	http://twitter.com	España
VETERINARIO/A¶#trabajo #empleo #Murcia	Tweet	2010-04-14 11:07:17	2019-05-13 08:59:42	Buffer	https://buffer.com	Murcia (España)
RT @PipiEstrada1: Pipíldoras: El futuro de los	Retweet	2010-02-09 11:43:20	2019-05-13 08:59:25	Twitter Web App	https://mobile.twitter.com	España
UGT agradece a todos los trabajadores y tral	Tweet	2009-10-30 19:26:44	2019-05-13 08:59:17	Twitter Web Clie	http://twitter.com	España
RT @JDamiann: Godín ya está en el santoral)	Retweet	2010-02-09 11:43:20	2019-05-13 08:59:10	Twitter Web App	https://mobile.twitter.com	España
#PymesUnidas Baúl de Algodón @bauldealg	Tweet	2014-05-17 08:41:17	2019-05-13 08:58:45	Facebook	http://www.facebook.com	España
Premarcos de madera: principales sistemas	Tweet	2011-04-15 15:00:10	2019-05-13 08:58:21	ndp_economia	http://www.comunicae.cc	Barcelona
El Puerto de Santander vuelve a ser punto de	Tweet	2012-03-24 19:08:26	2019-05-13 08:58:00	TweetDeck	https://about.twitter.com	Toledo
Gabilondo quiere renegociar la deuda de la C	Tweet	2010-03-09 10:05:13	2019-05-13 08:57:49	TweetDeck	https://about.twitter.com	Murcia, Spain
RT @AliciaPomares: Madre mía, que ilusión, !	Retweet	2009-05-10 17:24:11	2019-05-13 08:57:26	Twitter for Andr	http://twitter.com/downlc	Sant Cugat, Barcelor
#Abengoa construirá una desaladora en Emi	Tweet	2008-05-29 09:10:57	2019-05-13 08:57:24	Buffer	https://buffer.com	España
RT @thyssenkruppES: Te acompañamos en la	Retweet	2014-05-17 08:41:17	2019-05-13 08:57:04	Twitter Web Clie	http://twitter.com	España
RT @MiEconomista: 3 ventajas de la nueva ve	Retweet	2014-05-17 08:41:17	2019-05-13 08:57:01	Twitter Web Clie	http://twitter.com	España
RT @zonabienestar: Ayudar a los hijos en el p	Retweet	2014-05-17 08:41:17	2019-05-13 08:56:59	Twitter Web Clie	http://twitter.com	España
RT @bolsosmonai: #Monedero para tu día a	Retweet	2014-05-17 08:41:17	2019-05-13 08:56:57	Twitter Web Clie	http://twitter.com	España
RT @JFInformtica: https://t.co/oxAYqt072D Gu	Retweet	2014-05-17 08:41:17	2019-05-13 08:55:40	Twitter Web Clie	http://twitter.com	España
RT @JFInformtica: #pymesunidas #pymes #e	Retweet	2014-05-17 08:41:17	2019-05-13 08:55:07	Twitter Web Clie	http://twitter.com	España
Los riesgos relacionados con las TICs cent	Tweet	2012-06-19 06:15:44	2019-05-13 08:55:05	Hootsuite Inc.	https://www.hootsuite.co	Valladolid

Figura 6: Ejemplo de registro del modelo para los *tweets* en BD Curricula

Preprocesado

El preprocesado de los datos conforma hoy día una de las etapas más críticas dentro de los sistemas de aprendizaje automático al tener un impacto directo en la eficacia del sistema, ya que los mismos pueden contener ruido, estar incompletos o incoherentes.

En muchas ocasiones resulta difícil lograr una buena calidad de los datos de entrada, y dada la complejidad que esto supone se ha dividido el preprocesado en las siguientes sub-etapas:

Escalado y normalizado

Se trata de que todos los atributos trabajen sobre la misma escala y proporción, o lo que es lo mismo, se necesita comprimir o extender los valores de la variable para que estén en un rango definido.

Para el caso de los *tweets* almacenados en la base de datos, una vez se eliminaron los caracteres extraños, símbolos, *stickers*, saltos de línea, uso de letras repetidas; con el objetivo de obtener los *tweets* que contengan información útil, que ayude a clasificar los conjuntos de datos de forma no supervisada, utilizamos la herramienta TextBlob[27] (de la que se hablará más adelante).

El algoritmo *NaiveBayesAnalyzer()*[28] que el propio TextBlob nos proporciona se encarga de estandarizar los valores en *positivo*, *negativo* o *neutral* usando un escalado *MinMax*[29].

NaiveBayesAnalyzer() es una implementación del módulo *textblob.sentiments*, dicha implementación contiene un clasificador NLTK[30] que a su vez devuelve una tupla de la forma: *Sentiment(classification, p_pos, p_neg)* donde *classification* es el valor del sentimiento, *p_pos* y *p_neg* el rango de valores normalizados entre [0,1].

MinMax por su parte es uno de los método más simple dentro del escalado y consiste en volver a escalar el rango de características en [0, 1] o [-1, 1]. La fórmula general se muestra en 4.

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{4}$$

Donde x' es el valor escalado y x el valor a escalar, $min(x)$ es mínimo valor de x y $max(x)$ el máximo.

Es válido aclarar que con una sola implementación de *NaiveBayesAnalyzer()* se puede hacer una doble clasificación. Antes de dar un valor final, se puede clasificar una primera vez teniendo en cuenta las propiedades: *polaridad* y *subjetividad* que devuelve la función *getSentiment()*. La primera es un valor que se encuentra en el rango de $[-1, 1]$. Las oraciones subjetivas generalmente se refieren a la opinión personal, la emoción o el juicio; esta es también un valor numérico que se encuentra en el rango de $[0, 1]$.

Posteriormente se realiza una segunda clasificación, dando opción de valorar, cual de las dos es más cercana a la realidad según el contenido y el contexto en que se encuentra el mismo. Tal como se muestra en la Figura 7.

[27] *https://textblob.readthedocs.io/en/dev/*
[28] *https://textblob.readthedocs.io/en/dev*
[29] *https://scikit-learn.org/stable/modules/preprocessing.html#standardization-or-mean-removal-and-variance-scaling*
[30] *http://www.nltk.org*

```
INFO: Sentiments Analize
INFO: Tweet: Hoy celebramos los V edición de los de LA RAZÓN ➪Lo podrás ver en directo en nuestro canal
INFO: Polaridad: 0.0 Sentimiento: neutral Subjetividad: 0.0
INFO: Naives Bayes Analize
INFO: Sentiment(classification='pos', p_pos=0.5374999999999995, p_neg=0.4625)
INFO: Clasificado: pos
```

Figura 7: Ejemplo de *tweet* clasificado con TextBlob teniendo en cuenta la *polaridad* y la *subjetividad* y una clasificación general

Una vez escalados y normalizados los valores númericos del *tweet* a clasificar, se almacenan dichos valores en el *modelo tweet* de la base de datos para su posterior tratamiento (Ver Figura 8).

ABC val_feeling	123 val_pos	123 val_neg	123 val_neu
neg	0,452884097	0,547115903	0,0942318059
neg	0,3538218816	0,6461781184	0,2923562367
neg	0,3896325744	0,6103674256	0,2207348513
pos	0,9519917666	0,0480082334	0,0000
pos	0,5000	0,5000	0,0000
pos	0,5545734051	0,4454265949	0,0000
pos	0,8641304348	0,1358695652	0,0000
pos	0,5375	0,4625	0,0000
neg	0,4897864829	0,5102135171	0,0204270343
pos	0,5000	0,5000	0,0000
neg	0,4257322176	0,5742677824	0,1485355649
pos	0,5000	0,5000	0,0000
neg	0,2406716418	0,7593283582	0,5186567164
neg	0,3500	0,6500	0,3000

Figura 8: Sentimiento (val_feeling), valores numéricos positivos (val_pos), negativos (val_neg) y neutrales (val_neu) escalados y normalizados

Para el escalado de los valores antes descritos sólo se requirió TextBlob. Pero para los valores *tipo de tweet (val_tweet)* y los contextuales como: *horario (val_time)*, *año (val_year)*, *tipo de persona (val_person)* y *localización (val_location)* los cuales se muestra en la Figura 9, primeramente se requirió estandarizar dichos valores con un preprocesamiento manual teniendo en cuenta los valores que se muestran en la Tabla 2 y posteriomente se compararon dos tipos de escalados, uno basado en el *MinMax* y el otro en la desviación estándar, optando finalmente por el método de la desviación estándar; dichos resultados se muestran en la Figura 10. De esta forma se identifican los valores atípicos o diferentes en el conjunto de datos.

La desviación estándar es la medida de dispersión más común, que indica qué tan dispersos están los datos con respecto a la media. Mientras mayor sea la desviación estándar, mayor será la dispersión de los datos. Está descrita por la fórmula 5:

$$x' = \frac{x - \bar{x}}{\sigma} \tag{5}$$

El símbolo σ (sigma) se utiliza frecuentemente para representar la desviación estándar de una población, x representa la población, mientras que x' se utiliza para representar la desviación estándar de una muestra.

Campo	Forma númerica
val_tweet	0- tweet, 1- retweet
val_time	0- mañana, 1- mediodia, 2- tarde, 3- noche
val_year	0- año 2018, 1- año 2019, 2- año 2020, 3 - del 2021 en adelante
val_person	0- persona física, 1- persona jurídica
val_location	0- Andalucía, 1- Aragón, 2- Cantabria, 3- Castilla y León, 4- Cataluña, 5- Valencia, 6- Galicia, 7- La Rioja, 8- Ma- drid, 9- Murcia, 10- Navarra, 11- País Vasco, 12- Castilla la mancha, 13- España, 14- Asturias, 15- Sin clasificar

Tabla 2: Preprocesamiento manual para campos de textos a numéricos

val_tweet	val_time	val_year	val_person	val_location
0	0	2	1	15
0	0	2	1	15
0	0	2	1	14
0	0	2	1	15
0	3	1	1	13
0	0	2	1	15
0	0	2	1	15
1	0	2	1	14

Figura 9: Valores *tipo de tweet*, *horario*, *año*, *tipo de persona* y *localización* antes del escalado y normalizado

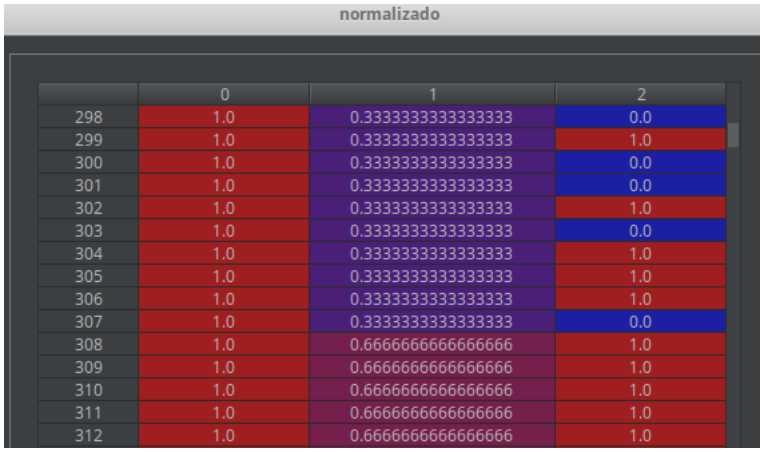
Reducción de la dimensionalidad

En los problemas de clasificación, una de las tareas a la que nos enfrentamos es seleccionar aquellos atributos que mejor describen la variable objetivo. Esto es, elegir aquellos atributos que aportan más información y tienen mayor correlación con la variable explicada [47].

Se trata de realizar una selección de los atributos computables dentro del modelo. Se busca descartar correlaciones y redundancias así como eliminar el ruido, reduciendo la dimensionalidad para que el modelo responda eficientemente (con certeza y velocidad).

Al tratarse de un *sistema no supervisado*, se opta por utilizar la medida de la entropía de cada atributo de tal forma que se pueda obtener mayor variabilidad.

La entropía es una medida del desorden (incertidumbre o dispersión) de un con-



	0	1	2
298	1.0	0.3333333333333333	0.0
299	1.0	0.3333333333333333	1.0
300	1.0	0.3333333333333333	0.0
301	1.0	0.3333333333333333	0.0
302	1.0	0.3333333333333333	1.0
303	1.0	0.3333333333333333	0.0
304	1.0	0.3333333333333333	1.0
305	1.0	0.3333333333333333	1.0
306	1.0	0.3333333333333333	1.0
307	1.0	0.3333333333333333	0.0
308	1.0	0.6666666666666666	1.0
309	1.0	0.6666666666666666	1.0
310	1.0	0.6666666666666666	1.0
311	1.0	0.6666666666666666	1.0
312	1.0	0.6666666666666666	1.0

Figura 10: Escalado, normalizado de los valores *val_time*, *val_person* y *val_location*

junto datos [47], y se define como la Fórmula 6:

$$H(X) = \sum_{i}^{n} p_i * log_2(p_i) \tag{6}$$

Donde p_i es la probabilidad relativa de aparición de la propiedad i en el conjunto de datos.

El resultado final, nos indica la medida de dispersión de la variable explicada con un valor que va de 0 a 1. Un valor de 0 indica orden total, o lo que es lo mismo, o todos varían o todos no lo hacen, pero todos los valores son iguales. Cuanto más cerca esté de 1, mayor desorden. Un valor 1 querría decir que el 50 % varían y el otro 50 % no.

En la Figura 11 de los cinco valores analizados, se puede apreciar que *val_year* al resultar ser 0 no se tuvo en cuenta porque influiría en el modelo, enmascarando otros datos importantes y por ende se desecha.

```
Value entropy val_tweet: 0.36564225626574726
Value entropy val_time: 0.7992866001937609
Value entropy val_year: 0.0
Value entropy val_person: 0.359430947137323
Value entropy val_location: 0.13626989970916922
unorganized values  [0.36564225626574726, 0.7992866001937609, 0.0, 0.359430947137323, 0.13626989970916922]
organized values  [0.0, 0.13626989970916922, 0.359430947137323, 0.36564225626574726, 0.7992866001937609]
final values  [0.7992866001937609, 0.359430947137323, 0.13626989970916922]
columns to keep  ['val_location', 'val_time', 'val_person']
```

Figura 11: Ejemplo cálculo de entropía

Aprendizaje

Las redes auto-organizadas deben descubrir rasgos comunes, regularidades, correlaciones o categorías en los datos de entrada, e incorporarlos a su estructura interna de conexiones [41]. Se dice, por tanto, que las neuronas deben auto-organizarse en función de los estímulos (datos) procedentes del exterior.

Un modelo SOM está compuesto por dos capas de neuronas. La capa de entrada (formada por N neuronas, una por cada variable de entrada) se encarga de recibir y transmitir a la capa de salida la información procedente del exterior. La capa de salida (formada por M neuronas) es la encargada de procesar la información y formar el mapa de rasgos [45]. Normalmente, las neuronas de la capa de salida se organizan en forma de mapa bidimensional como se muestra en la Figura 12.

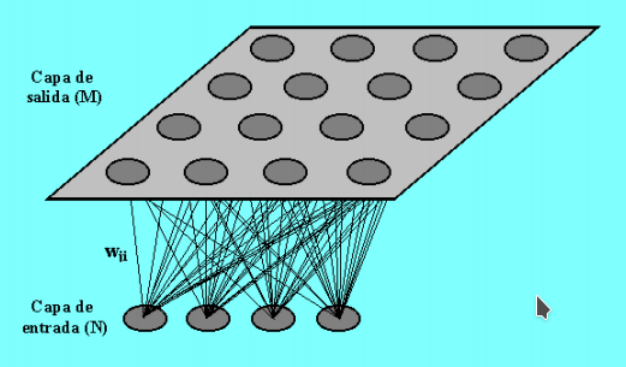


Figura 12: Arquitectura del SOM: tomada de [45]

Las conexiones entre las dos capas que forman la red son siempre hacia delante, es decir, la información se propaga desde la capa de entrada hacia la capa de salida. Cada neurona de entrada i está conectada con cada una de las neuronas de salida j mediante un peso Wji. De esta forma, las neuronas de salida tienen asociado un vector de pesos Wj llamado vector de referencia (o *codebook*), debido a que constituye el vector prototipo (o promedio) de la categoría representada por la neurona de salida j. Así, el SOM define una proyección desde un espacio de datos multidimensional a un mapa bidimensional de neuronas [45].

Entre las neuronas de la capa de salida, puede decirse que existen conexiones laterales de excitación e inhibición implícitas, pues aunque no estén conectadas, cada una de estas neuronas va a tener cierta influencia sobre sus vecinas. Esto se consigue a través de un proceso de competición entre las neuronas y de la aplicación de una función denominada de vecindad, que produce la topología o estructura del mapa. Las topologías más frecuentes son la rectangular y la hexagonal.

Las neuronas adyacentes pertenecen a una vecindad Nj de la neurona j. La topología y el número de neuronas permanece fijo desde el principio [45]. El número de neuronas determina la suavidad de la proyección, lo cual influye en el ajuste y capacidad de generalización del SOM.

Durante el entrenamiento, el SOM forma una red elástica que se pliega dentro de la nube de datos originales. El algoritmo controla la red de modo que tiende a aproximar la densidad de los datos. Los vectores de referencia del *codebook* se acercan a las áreas donde la densidad de datos es alta [45].

En el aprendizaje competitivo las neuronas compiten unas con otras con el fin de llevar a cabo una tarea dada. Se pretende que cuando se presente a la red un patrón de entrada, sólo una de las neuronas de salida (o un grupo de vecinas) se active. Por

tanto, las neuronas compiten por activarse, quedando finalmente una como neurona vencedora y anuladas el resto, que son forzadas a sus valores de respuesta mínimos (Ver Figura 13).

El objetivo de este aprendizaje es provocar que diferentes partes de la red respondan similarmente a ciertos patrones de la entrada. Y a la misma vez, categorizar los datos que se introducen en la red. Se clasifican valores similares en la misma categoría y, por tanto, deben activar la misma neurona de salida.

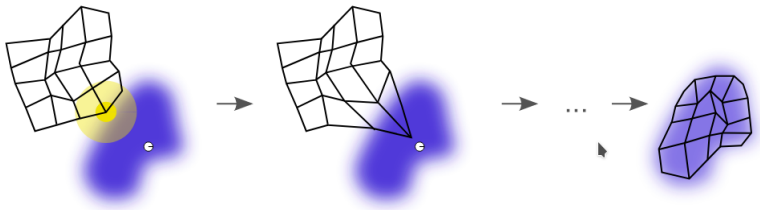


Figura 13: Entrenamiento de un mapa auto-organizado. La zona coloreada es la distribución de los datos de entrenamiento, y la red neuronal es el ejemplo de entrada actual para esa distribución: tomada de Wikipedia [48]

El proceso de aprendizaje del SOM es el siguiente:

Un vector x es seleccionado al azar del conjunto de datos y se calcula su distancia (similitud) a los vectores del *codebook*, usando, por ejemplo, la distancia euclidiana.

Una vez que se ha encontrado el vector más próximo o *Best Matching Unit* (BMU) el resto de vectores del *codebook* son actualizados. El BMU y sus vecinos (en sentido topológico) se mueven cerca del vector x en el espacio de datos. La diferencia entre ambos valores se decrementan con el tiempo, así como la distancia desde el BMU. La Fórmula 7 se emplea para actualizar una neurona con su vector de peso.

$$W_v(s+1) = W_v(s) + \theta(u, v, s)\alpha(s)(D(t) + W_v) \tag{7}$$

donde s es el número de ciclo, t es el índice dentro del conjunto de entrenamiento, u es el índice de BMU para el vector de entrada, v es el índice de la siguiente entrada, y $\alpha(s)$ es el coeficiente monótonamente decreciente de aprendizaje. [42]

La función de vecindad $\theta(u,v,s)$ depende de la distancia de cuadrículas entre la BMU (neurona u) y la neurona v. De forma simple se le da el valor 1 a todas las neuronas suficientemente cerca a BMU y 0 a las otras, pero es más común elegir una función gaussiana. Independientemente de la forma funcional, la función de vecindad se contrae con el tiempo [40]. La magnitud de dicha atracción está regida por la *tasa de aprendizaje*.

Al inicio, la auto-organización tiene lugar a escala global (Ver Figura 14). Cuando la vecindad ha sido ajustada a solo unas cuantas neuronas, los pesos irán convergiendo a estimaciones locales, reduciendo así el número de iteraciones y el error de cuantificación (Ver Figura 15).

En algunas implementaciones, el coeficiente de aprendizaje, α , y la función de

```
Finetune training...
radius_ini: 5.833333 , radius_final: 1.000000, trainlen: 26

epoch: 1 ---> elapsed time:  1.609000, quantization error: 0.145931

epoch: 2 ---> elapsed time:  1.517000, quantization error: 0.252822

epoch: 3 ---> elapsed time:  1.619000, quantization error: 0.225133

epoch: 4 ---> elapsed time:  1.519000, quantization error: 0.208946

epoch: 5 ---> elapsed time:  1.614000, quantization error: 0.195810
```

Figura 14: Número de iteraciones, tiempo transcurrido y error de cuantificación, al inicio del entrenamiento

vecindad, θ, decrecen de manera constante con el incremento de s, en otras (en particular aquellas donde t explora rápidamente el conjunto de entrenamiento) el decrecimiento ocurre más lentamente.

Este proceso es repetido para cada vector de entrada un número de ciclos, λ, usualmente grande, hasta que el entrenamiento termina. La red va asociando las neuronas de salida con grupos o patrones en el conjunto de entrenamiento. El número de pasos de entrenamiento se debe fijar antes a priori, para calcular la tasa de convergencia de la función de vecindad y de la tasa de aprendizaje.

Una vez terminado el entrenamiento, el mapa ha de ordenarse en sentido topológico: Los n vectores topológicamente próximos se agrupan en n neuronas adyacentes o incluso en la misma neurona.

Durante el mapeo, solo existirá una neurona ganadora, la neurona cuyo vector de pesos se encuentre más cerca del vector de entrada.

```
epoch: 22 ---> elapsed time:  1.545000, quantization error: 0.030212

epoch: 23 ---> elapsed time:  1.632000, quantization error: 0.024866

epoch: 24 ---> elapsed time:  1.574000, quantization error: 0.020676

epoch: 25 ---> elapsed time:  1.622000, quantization error: 0.016940

epoch: 26 ---> elapsed time:  1.513000, quantization error: 0.013688

Final quantization error: 0.013688
train took: 67.405000 seconds
```

Figura 15: Número de iteraciones, tiempo transcurrido y error de cuantificación, finalizado el entrenamiento

Durante el entrenamiento para el conjunto de entradas *Negative*, *Positive*, *Neutral*, *Location*, *Hour* y *Type Person*, con los valores establecidos en la arquitectura, SOMPY estableció un radius_ini (radio de inicio) de 5,833333, un radius_final de 1,000000 y un *trainlen* (entrenamiento) igual a 26. Luego la red neuronal clasifica (agrupa) topológicamente las entradas transformándolas en patrones o clases, de tal forma que la probabilidad de error para el modelo esté por debajo de 0.013688 (Ver Figura 15).

Por lo que podemos concluir, que al final del entrenamiento, la red neuronal clasifica (agrupa) topológicamente las entradas transformándolas en patrones o clases.

Evaluación y Clasificación

SOM trata de encontrar una proyección no lineal óptima para los datos multidimensionales, de manera que los vectores que se proyectan en la superficie bidimensional, conservan la misma distancia euclídea relativa entre ellos.

Dichos mapas resultantes sirven para dos cosas:

1. Descubrir las relaciones entre las variables *negative*, *positive*, *neutral* y el resto de variables contextuales de mayor entropía, en este caso: *location*, *hour*, *type person*.

Cuando están correlacionadas, es sencillo verlas. Por ejemplo, en la Figura 16 las variables *negative* y *positive* son el inverso unas de otras (es el mismo con otro color).

2. Para hacer la clasificación, ya que a través de los mapas y el modelo SOM, cuando recibimos una nueva entrada se podrá clasificar a que grupo pertenece.

Una vez que se ha entrenado el mapa, es importante saber si se ha adaptado adecuadamente a los datos de entrenamiento. Como medidas de calidad de los mapas se considera la precisión de la proyección y la preservación de la topología.

La medida de precisión de la proyección describe cómo se adaptan o responden las neuronas a los datos. Habitualmente, el número de datos es mayor que el número de neuronas y el error de precisión es siempre diferente de 0.

La medida de preservación de la topología por su parte describe la manera en la que el SOM preserva la topología del conjunto de datos. Esta medida considera la estructura del mapa.

Una manera simple de calcular el error topográfico está dado por la Fórmula 8:

$$\varepsilon_t = \frac{1}{N} \sum_{k=1}^{N} u(x_k) \tag{8}$$

donde $u(x_k)$ es igual a 1 si el primer y segundo BMUs de x_k no están próximos el uno al otro. De otro modo, $u(x_k)$ es igual a 0.

Cada instanciación de estos parámetros conducirá a un modelo diferente. Para elegirlos necesitamos un criterio para evaluar la calidad del aprendizaje. Este criterio puede depender realmente de la tarea que estamos resolviendo con el SOM. Sin embargo, un enfoque general es observar el error de cuantificación, que se muestra en la Figura 15.

Los pesos de las neuronas son como punteros al espacio de entrada. Estos forman una aproximación discreta a la distribución de las entradas de entrenamiento. Se producen más punteros a regiones de alta concentración y menos donde son más

escasas.

Después del entrenamiento, los Mapas auto-organizados de las Figuras 16 y 17 muestran los patrones de comportamiento de las personas relacionadas con Faster en Twitter, y de las variables contextuales.

En la Figura 16 se representa el mapa resultante para las entradas *Negative*, *Positive* y *Neutral* y su relación con los valores restantes. Se puede apreciar además que el sentimiento positivo y negativo predomina por encima de los neutrales; por lo tanto puede que en algún momento no se tengan en cuenta los neutrales en el modelo, o que se realice alguna tarea de ML para tratar que se conviertan en positivos o negativos.

Por su parte la Figura 17 muestra los mapas para las entradas *Location*, *Hour* y *Type person*. Se observa como la localización no varía mucho, y esto se debe a que muchas personas no especifican al detalle el lugar exacto del cual están publicando en Twitter, puede que sea necesario realizar segmentación por provincias en lugar de agrupar por Comunidad Autónoma. La hora de publicación por el contrario si es muy variable, algo que no pasa con el tipo de persona (física y jurídica), ya que al ser de dos tipos específicos no se salen de su valor estándar y muestran resultados más homogéneo comparados con el resto de variables.

Figura 16: Mapas SOMPY para las entradas negative, positive, neutral

En los mapas resultantes se observan, que existen zonas en que la distancia promedio es alta, y por ende los pesos circundantes son muy diferentes; a dichas zonas se le asigna un color claro. En las zonas en que la distancia promedio es baja, se asigna un color más oscuro. Las Figuras 16 y 17 nos dicen además, dónde la densidad de las personas es mayor (regiones más oscuras) o menor (regiones más claras).

Si observamos el error de cuantificación, que para dichos mapas es igual a 0,013688 se puede decir que el modelo ha sido capaz de aprender y relacionar los datos de entrada y los resultados. Puede inferirse además que al no ser un modelo complejo, tiene mayor capacidad de división para poder realizar un buen aprendizaje y por

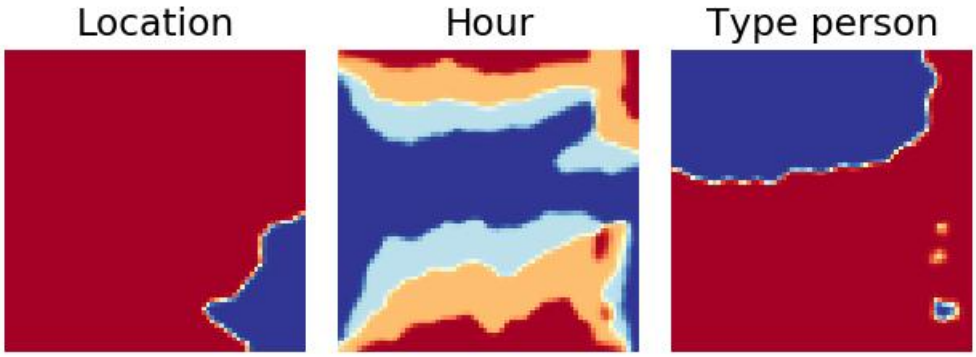


Figura 17: Mapas SOMPY para las entradas location, hour, type person

ende puede preservar mejor la topología.

Debido a que en la fase de entrenamiento los pesos de toda la vecindad son movidos en la misma dirección, elementos similares tienden a excitar neuronas adyacentes. Por tanto, forman un mapa semántico donde entradas similares son mapeadas juntas, hecho muy importante en el procesamiento de los mensajes textuales [43].

Herramientas utilizadas

Las herramientas empleadas en el desarrollo de este trabajo se expresan brevemente en la tabla 3

Herramienta	Descripción	Versión
Python	Lenguaje de programación principal.	v 2.7
Tweepy	Biblioteca de Python para interacción con API de Twitter.	v 3.5.0
TextBlob	Plataforma para NPL.	v 0.15.3
Scikit-Learn	Biblioteca de Python para trabajo con ML.	v 0.20.3
SOMPY	Biblioteca de Python para SOM con entrenamiento por lotes.	v 1.0
Pandas	Biblioteca de Python para el análisis de datos.	v 0.24.2
PostgreSQL	Sistema gestor de bases para almacenar conjunto de datos.	v 10.9
Flask	Microframework para Python para Aplicaciones Web.	v 1.0.2
Angular	Framework MVC de JavaScript para Desarrollo FrontEnd.	v 7.3.8

Tabla 3: Herramientas utilizadas durante el desarrollo

Python

Python es un lenguaje de programación interpretado de alto nivel, cuya filosofía está orientada a la legibilidad del código, que debe contar con una sintaxis clara y expresiva. Se caracteriza por disponer de una amplia librería estándar y hacer que la tarea de programación sea más rápida y productiva, por requerir menos tiempo de ejecución, consumo de memoria, líneas de código y esfuerzo de desarrollo, comparado con el mismo programa escrito en otros lenguajes como C, C++ o Java. [29]

Es un lenguaje de programación versátil multiplataforma y multiparadigma que se destaca por su código legible y limpio. La licencia de código abierto permite su utilización en distintos contextos sin la necesidad de abonar por ello y se emplea en plataformas de alto tráfico como Google, YouTube o Facebook. [30] Principalmente es un lenguaje orientado a objetos, todo en Python es un objeto, pero también incorpora aspectos de la programación imperativa, funcional, procedural y reflexiva. Su objetivo es la automatización de procesos para ahorrar tanto complicaciones como tiempo, los dos pilares en cualquier tarea profesional. Dichos procesos se reducirán en pocas líneas de código que insertarás en una variedad de plataformas y sistemas operativos. [30]

Fue concebido al inicio de los años 90 por Guido van Rossum, pero no se hizo popular hasta el lanzamiento de su segunda versión en el año 2000, cuando su desarrollo se hizo disponible a la comunidad en los términos de los Python Enhancement Proposals[31], documento que define características y mejoras del lenguaje. Enla actualidad, se encuentra en su tercera versión, también conocida por Python 3.0, un lanzamiento importante, que se caracterizó por la incompatibilidad con las versiones anteriores, con el objetivo de corregir los fallos que se han descubierto y para limpiar los excesos de las versiones anteriores. [31]

Gracias a sus potentes funciones es ideal para trabajar con grandes volúmenes de datos porque favorece su extracción y procesamiento, siendo el elegido por las empresas de Big Data. A nivel científico, posee una amplia biblioteca de recursos con especial énfasis en las matemáticas para aspirantes a programadores en áreas especializadas. [30]

Tweepy

Tweepy es una biblioteca desarrollada en Python que provee una interacción con la API oficial de Twitter. Funciona además como una capa abstracta que se comunica con el API REST de Twitter encapsulando las peticiones como una simple función en Python; de esta forma el desarrollador no tiene que preocuparse con el resto de capas del módulo.

La API proporciona acceso a todos los métodos de la API RESTful de twitter. Cada método puede aceptar varios parámetros y devolver respuestas. Cuando invocamos un método API, la mayor parte del tiempo nos devolverá una instancia de

[31] *https://www.python.org/dev/peps/*

clase de modelo Tweepy. Esto contendrá los datos devueltos de Twitter que luego podemos usar dentro de nuestra aplicación.

A través de la misma se puede sacar partido de las funcionalidades que la API ofrece en aplicaciones desarrolladas por terceros, como la consulta de *tweets* públicos en tiempo real, mediante la *Streaming API*[32] o de *tweets* públicos recientes o populares publicados en los últimos 7 días, a través de la Search API[33], o trabajar con los métodos de líneas de tiempo (Timeline[34]) donde puedes obtener entre otras los 20 estados más recientes, incluidos los retweets, publicados por el usuario autenticado y los amigos de ese usuario.

Tweepy inclusive ofrece el soporte integrado a la forma de autenticación a las llamadas de la API de Twitter hechas mediante OAuth[35], que es un protocolo seguro de autorización que permite terceras aplicaciones acceder a información de un usuario sin que estos conozcan sus credenciales de autenticación.

Con tan solo registrar estas aplicaciones en el portal de desarrolladores de Twitter, los usuarios están dando su permiso a las aplicaciones a que puedan acceder a las funciones de su cuenta. Una vez autorizada, Twitter suministra 2 claves: la access token key y access token secret, que le concede a la aplicación el permiso para realizar acciones en la red social actuando en nombre del usuario sin pedir contraseña.

TextBlob

TextBlob[36] es una biblioteca de Python para procesar datos de texto. Proporciona una API consistente para sumergirse en tareas comunes de procesamiento de lenguaje natural (NLP) tales como etiquetado de parte del habla, extracción de frases sustantivas, análisis de sentimientos, entre otras.

TextBlob no sólo realiza tareas de NLP sino que se basa en gran parte en *Natural Language Toolkit* (NLTK[37]), lo cual permite que pueda disponerse del conjunto de herramientas que NLTK ofrece y de las suyas propias, como son:

- Analizador semántico
- Analizador morfológico (*Stemming* y *Lematizador*)
- Analizador sintácticos (*Parsing*)
- Etiquetado Gramatical (*POS Tagging*)
- Segmentador de palabras, oraciones y párrafos (*Tokenizer*)

[32] *https://developer.twitter.com/en/docs/tutorials/consuming-streaming-data.html*
[33] *https://developer.twitter.com/en/docs/tweets/search/overview/standard*
[34] *https://developer.twitter.com/en/docs/tweets/timelines/overview.html*
[35] *https://developer.twitter.com/en/docs/basics/authentication/overview/oauth.html*
[36] *https://textblob.readthedocs.io/en/dev/index.html*
[37] *http://www.nltk.org*

- Modelo de n-grama
- Clasificación (Naive Bayes, Árbol de decisiones)
- Añadir nuevos modelos o idiomas a través de extensiones
- Extracción de frases
- Frecuencias de palabras y frases
- Traducción y detección de idiomas con Google Translate
- Inflexión de palabras (pluralización y singularización) y lematización
- Corrección ortográfica
- Integración y Soporte multi-idiomas (*WordNet*)

Scikit-Learn

Scikit-Learn[38] es una biblioteca que contiene una gama de algoritmos de aprendizaje supervisado y no supervisado para el lenguaje de programación Python. Este módulo se centra en presentar la máquina de aprendizaje automático a un público no especializado utilizando un lenguaje de alto nivel, para ello hace hincapié en la facilidad de uso, rendimiento, documentación y en una API consistente [32] y bien estructurada.

Tiene implementado algoritmos de clasificación, regresión, clustering, incluidos la SVM, bosques aleatorios (RF, *random forests*), gradientes estocástica(*stochastic gradient boosting*), *k-means* y hasta selección automática de modelos y análisis de resultados. Es de código abierto y es reutilizable en varios contextos, fomentando el uso académico y comercial. Está construida sobre SciPy[39] (Scientific Python) y presenta compatibilidad con otras bibliotecas numéricas y científicas de Python como NumPy[40], Pandas[41], Matplotlib[42], Ipy[43] y Sympy[44] [33].

El desarrollo del Scikit-learn se basa en herramientas colaborativas como Git[45], GitHub[46] y listas de distribución públicas para potenciar el uso compartido de la información. Las contribuciones externas son bienvenidas y alentadas. Pero, además del soporte de esa comunidad, se puede consultar una guía de usuario que incluye documentación, códigos, binarios, clases de referencia, tutoriales, manuales de instrucciones, así como ofrece más de 60 ejemplos de algunas aplicaciones del mundo real [32].

[38] *https://scikit-learn.org/stable/*
[39] *https://www.scipy.org/*
[40] *https://www.numpy.org/*
[41] *https://pandas.pydata.org/*
[42] *https://matplotlib.org/*
[43] *https://pypi.org/project/IPy/*
[44] *https://www.sympy.org/es/*
[45] *https://git-scm.com/*
[46] *https://github.com/*

La ventaja de la programación en Python, y Scikit-Learn en concreto, es la variedad de módulos y algoritmos que facilitan el aprendizaje y trabajo con ML. Por dichas razones, puede ser fácilmente integrado en aplicaciones más allá del ámbito de análisis de datos estadísticos.

SOMPY

SOMPY[47] es un paquete para Mapas Autoorganizados (SOM), que se basa en la implententación de un algoritmo de aprendizaje en el que se puede realizar entrenamiento por lotes, que además es rápido y escalable. Surge con la necesidad de acelerar el algoritmo que utiliza SOM y hacerlo adecuado para un tamaño de datos más grande.

Asociado con cada neurona hay un vector de pesos, de la misma dimensión de los vectores de entrada, y una posición en el mapa. La configuración usual de las neuronas es un espacio regular de dos dimensiones, en una rejilla hexagonal o rectangular. Los Mapas Autoorganizados describen un mapeo de un espacio de mayor dimensión a uno de menor dimensión. El procedimiento para ubicar un vector del espacio de los datos en el mapa es encontrar la neurona con el vector de pesos más cercano (menor distancia métrica) al vector del espacio de los datos.

Mientras que es típico considerar este tipo de estructura de la red de la misma familia que las redes con retro-alimentación, donde los nodos son visualizados como si estuvieran adheridos, este tipo de arquitectura es diferente en configuración y motivación.

Si bien, la mayoría de las aplicaciones de SOM se limitan a visualizaciones simples, hay que destacar que tiene muchas otras capacidades para el agrupamiento, clasificación, predicción, aproximación de funciones más allá del concepto de método de mínimos cuadrados y funciones polinómicas para el cálculo basado en características simbólicas (es decir, una función no lineal, contextual, que no tiene relaciones funcionales cerradas para su construcción de funciones de nivel inferior) e incluso para el análisis de datos multimodal [34].

Al ser una extensión de SOM, puede utilizar las mismas funcionalidades de este algoritmo como:

- Inicialización de *Principal Component Analysis* (PCA)[48] (o RandomPCA (predeterminado)), usando Scikit-Learn o inicialización aleatoria.
- Visualización Unidimensional (1-D) o Bidimensional (2-D), con una cuadrícula plana y rectangular.
- Diferentes métodos para la aproximación de funciones y predicciones (en su mayoría utilizando Scikit-Learn).

[47] *https://github.com/sevamoo/SOMPY*
[48] *https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html*

A partir de la versión actual del código, es tres veces más rápido que SomToolbox[49] de Matlab, aunque puede ser mejorable adaptando dicha biblioteca usando la computación paralela en un clúster [34].

Pandas

Pandas[50] es una biblioteca de código abierto de Python destinada al análisis de datos, que proporciona unas estructuras de datos flexibles y que permiten trabajar con ellos de forma muy eficiente [39]. Algo muy útil cuando se trabaja con aprendizaje automático en Python.

Proporciona además un conjunto de herramientas que permiten entre otras cosas:

- Estructuras de datos de Series y DataFrame.
- Leer y escribir datos en diferentes formatos: CSV, TSV, MS Excel.
- Variedad de utilidades para realizar operaciones de entrada/salida de manera transparente.
- Interfaz gráfica.

PostgreSQL

PostgreSQL[51] se caracteriza por ser un sistema estable, de alto rendimiento, gran flexibilidad ya que funcionar la mayoría de los sistemas Unix, además tiene características que permiten extender fácilmente el sistema. Permite desarrollar o migrar aplicaciones desde Access, Visual Basic, Foxpro, Visual Foxpro, C/C++ Visual C/C++, Delphi, etc., para que utilicen a PostgreSQL como servidor de BD; Por lo expuesto PostgreSQL se convierte en una gran alternativa al momento de decidirse por un sistema de bases de datos [35].

Está desarrollado desde 1996 por la comunidad partir del SGBD POSGRES, que surgió a partir de un proyecto de investigación militar estadounidense (DARPA, ARO) con participación civil [36]. Es un sistema gestor de bases de datos relacionales, está orientado a objetos (algo parecido a un lenguaje de programación) y multisistema (por tanto PostgreSQL puede ser instalado en Microsoft Windows, GNU/Linux, MacOS, BSD) y Open Source.

PostgreSQL es unos de los sistemas de gestión de bases de datos relacionales más usados en la actualidad. Es escalable y puede manejar bases de datos enormes, de más de 100 Terabytes [36].

Como características principales:

[49] *http://www.cis.hut.fi/somtoolbox/package/docs2/somtoolbox.html*
[50] *https://pandas.pydata.org/*
[51] *https://www.postgresql.org/*

- El lenguaje SQL que usa es muy próximo al estándar ISO/IEC, gracias a lo que resulta relativamente sencillo portar consultas y scripts de otros sistemas de bases de datos.
- Cumple con ACID, es decir provee atomicidad, consistencia, aislamiento y durabilidad para sus operaciones.
- Permite crear esquemas, tablas heredadas y triggers orientados a eventos que no poseen otros motores.
- Permite definir procedimientos, no solo en PostgreSQL, sino también en otros muchos lenguajes como Pearl, TCL o Python.
- Podemos extender la funcionalidad con extensiones, provistas por la propia PostgreSQL, por terceros o incluso programando por nuestra cuenta.
- Provee una excelente escalabilidad vertical.

Flask

Flask[52] es un "micro" *Framework* escrito en Python y concebido para facilitar el desarrollo de Aplicaciones Web bajo el patrón MVC [38].

Es un marco de aplicación web WSGI ligero. Está diseñado para que el inicio sea rápido y fácil, con la capacidad de escalar a aplicaciones complejas. Comenzó como un envoltorio simple alrededor de Werkzeug y Jinja y se ha convertido en uno de los marcos de aplicaciones web de Python más populares.

Flask ofrece sugerencias, pero no impone ninguna dependencia ni diseño del proyecto. Es responsabilidad del desarrollador elegir las herramientas y las bibliotecas que desean utilizar. La comunidad proporciona muchas extensiones que facilitan la adición de nuevas funciones [38].

Algunas de las características esenciales son:

- Incluye un servidor web de desarrollo.
- Tiene un depurador y soporte integrado para pruebas unitarias.
- Soporta de manera nativa el uso de cookies seguras.
- Sirve para construir servicios web (como APIs REST) o aplicaciones de contenido estático.
- Open Source y está amparado bajo una licencia BSD.
- Buena documentación, código de GitHub y una buena comunidad.
- Extensiones de Flask.

[52] *http://flask.pocoo.org/*

Flask es una alternativa para la construcción de Apps Web con Python, tiene una buena curva de aprendizaje y se puede aprender muy rápido.

Angular

Angular[53] es un framework de desarrollo para JavaScript creado por Google. La finalidad de Angular es facilitarnos el desarrollo de aplicaciones *Web Single-page Application* (SPA) y además darnos herramientas para trabajar con los elementos de una web de una manera más sencilla y óptima [37]. Cambia totalmente el desarrollo, ya que separa completamente el front-end y el back-end en una aplicación web.

Una aplicación web SPA creada con Angular es una web de una sola página, en la cual la navegación entre secciones y páginas de la aplicación, así como la carga de datos, se realiza de manera dinámica, casi instantánea, asincronamente haciendo llamadas al servidor (backend con un API REST) y sobre todo sin refrescar la página en ningún momento [37]. Son además reactivas y no recargan el navegador, todo es muy dinámico y asíncrono con ajax.

Ventajas o características:

- Aplicaciones modulares y escalares.
- Lenguaje Typescript, tiene una sintaxis muy parecida a Java, con tipado estático.
- Sigue el patrón MVC, con la vista separada de los controladores.
- Basado en componentes, es decir, podemos escribir componentes web con vista y lógica para después reutilizarlos en otras páginas.
- Inyección de dependencias, un patrón de diseño que se basa en pasar las dependencias directamente a los objetos en lugar de crearlas localmente.
- Programación reactiva, la vista se actualiza automáticamente a los cambios.
- Se integra bien con herramientas de testing.
- Se integra bien con Ionic, para adaptar aplicaciones web a dispositivos móviles.

Otra ventaja que tiene este framework es que está respaldado por Google y tiene una comunidad grande detrás.

Implementación del sistema

Como ya se ha comentado se pretende crear una herramienta que analice las opiniones de las personas que son seguidores de *@FasterEmpleo* en Twitter y que tienen

[53] *https://angular.io/*

alguna relación laboral con el *Grupo Faster*, para hacer un enfoque diferenciado a la tarea de análisis de sentimientos.

Se trata extraer información útil acerca del estado de ánimo de dichos seguidores, que pueda emplearse en la toma de decisiones de la empresa. Actualmente por ejemplo: Para detectar si el motivo del estado de ánimo negativo del usuario tiene que ver con su situación laboral, se realiza una encuesta de clima destacando aspectos como nivel de compriso con la empresa cliente, formación y/o motivación para tratar así de poder emprender acciones correctivas que mejoren su estado de ánimo en lo relativo a su desarrollo laboral.

Para el presente trabajo se optó por analizar la motivación del trabajador teniendo en cuenta las variables contextuales que nos aporta la base de datos del *Customer Relationship Management* (CRM) Odoo [54], la herramienta donde se gestionan la cartera de clientes y empleados del Grupo Faster. Dichas variables son: horas de trabajo (contract_hours), cantidad de días del contrato (contract_day) que pueden ser de *1* a *7 días*, salario (salary_real) y tipo de retribución (type_retribution) que puede ser *0 - Por horas*, *1- Por días* y *2 - Mensual*.

La herramienta empleará los métodos clásicos de clasificación de NLP, algunos algoritmos de ML para métodos no supervisados y la biblioteca de Mapas Autoorganizados SOMPY. La misma permitirá una intuitiva visualización de los resultados, a través de los mapas de clases resultantes. Para en base a ello, tomar acciones comerciales que ayuden a la mejora continua del Grupo.

Además se hace necesario una interfaz intuitiva, ágil y fácil de manejar para los usuarios finales; de forma que puedan interactuar o consultar parte o la totalidad de los datos; así como disponer de estadísticas de los mismos.

Siguiendo la metodología propuesta se procedió como se describe en la sección Extracción de datos a través de Tweepy a extraer los mensajes del *time_line* de Twitter del usuario *@FasterEmpleo*. Con un período de 20 minutos aproximadamente se extrae toda la información relacionada con los *tweets* y seguidores de Faster, para guardar en una base de datos de Postgres en: *Modelo para tweets*, *Modelo para usuarios* y *Modelo para Persona (Persona Física o Jurídica)*. En este último a su vez se recogen los datos contextuales (Ver Figura 18) que ayudarán a determinar la causa del sentimiento negativo en nuestros trabajadores.

Desde 2017 y hasta la fecha actual se han recopilado un total de 302,772 *tweets*, de los cuales 90,258 fueron clasificados como negativos a través del análisis de sentimiento que se realizó en la sección Escalado y normalizado lo cual representan el 29,8 % del total.

De las 657 personas registradas en nuestra base de datos se conoce que 403 son personas físicas, y 251 están identificados como empleados nuestros, lo que representa un 2,28 % por encima de la media.

Para este grupo de empleados se tomaron los valores contextuales del *Modelo Persona* (Ver Figura 18) y a través de algunas técnicas de ML y la biblioteca

[54] *https://odoo10.faster.es/*

SOMPY se procedió a analizar los posibles factores que influyen en su estado de ánimo negativo.

contract_hours	contract_day	salary_real	type_retribution
20	3	8,3000	1
30	4	9,9700	2
30	4	6,9400	2
30	4	7,1200	1
30	4	9,5000	1
20	2	6,9800	1
20	2	6,9800	1
16	2	6,9800	1
16	2	6,9800	1
40	5	13,1000	1

Figura 18: Valores *contract_hours*, *contract_day*, *salary_real* y *type_retribution*

De cara a la clasificación se procedió a realizar el Preprocesado de los datos del *tweet*, la asociación de los datos contextuales con el mismo y los del grupo de empleados; para ello se llevaron todos los atributos a la misma escala y proporción, y se redujo la dimensionalidad.

A la vez y en paralelo, con el objetivo de aminorar las transaciones a la base de datos, haciendo uso de la librería Pandas procedemos a distribuir los valores en dos ficheros *.csv*:

En el primer fichero *classification.csv* se almacenan los datos relacionados con el *tweet* (Ver la Figura 19) más los valores como: tipo de tweet (*val_tweet*) y los contextuales como: horario (*val_time*), año (*val_year*), tipo de persona (*val_person*) y localización (*val_location*).

Una vez identificados los *tweets* negativos y el usuario asociado al mismo en el fichero *classification.csv*, en un segundo fichero llamado *classificationemployee.csv* se almacenan los datos relacionados con el usuario identificado como empleado del *Grupo Faster*, los campos almacenados son: *contract_hours*, *contract_day*, *salary_real* y *type_retribution* (Ver Figura 20).

```
user_id,p_pos,p_neg,p_neu,val_tweet,val_time,val_year,val_person,val_location
277901,0.2228915662650598,0.7771084337349403,0.0,1,0,2,0,15
277902,0.28260869565217395,0.7173913043478259,0.0,0,0,2,1,15
277903,0.5,0.5,0.0,0,0,2,0,15
277904,0.6507352941176469,0.34926470588235325,0.3014705882352936,0,0,2,1,15
277905,0.6224489795918364,0.3775510204081633,0.24489795918367313,0,0,2,1,15
277906,0.2499999999999997,0.75,0.0,0,0,2,0,15
277907,0.3541666666666662,0.6458333333333348,0.0,0,0,2,0,15
277908,0.2873242749582872,0.7126757250417122,0.0,0,0,2,1,15
277909,0.5,0.5,0.0,0,0,2,0,15
277910,0.5,0.5,0.0,0,0,2,1,15
```

Figura 19: Ejemplo de valores almacenados en el fichero *classification.csv*

Para el Escalado y normalizado de las variables contextuales los valores *contract_hours*, *contract_day*, *salary_real* y *type_retribution* al contrario de ejemplos anteriores se utilizó la técnica *MinMax* evalúandolo en un rango [0, 1], quedando la normalización según se muestra en la Figura 21.

```
person_id, contract_hours, contract_day, salary_real, type_retribution
17,20,2,6.98,1
200,40,5,11.9,1
380,40,5,6.94,2
20,20,5,13.1,1
325,20,3,7.67,2
327,20,3,9.5,2
372,20,3,6.94,2
454,20,5,6.94,2
329,20,3,9.46,2
464,20,5,6.94,2
```

Figura 20: Ejemplo de valores almacenados en el fichero *classificationemployee.csv*

normalizado

0	1	2	3
0.5	0.9999999999999999	0.16666666666666663	1.0
1.0	0.33333333333333337	0.16666666666666663	0.11850649350649345
1.0	0.33333333333333337	0.16666666666666663	0.4155844155844155
1.0	0.33333333333333337	0.16666666666666663	0.0
1.0	0.9999999999999999	0.16666666666666663	0.0
1.0	0.33333333333333337	0.16666666666666663	0.40909090909090917
1.0	0.9999999999999999	0.16666666666666663	0.0
1.0	0.9999999999999999	0.16666666666666663	0.0
0.5	0.33333333333333337	0.16666666666666663	0.06655844155844148
1.0	0.33333333333333337	0.16666666666666663	0.09090909090909083
1.0	0.33333333333333337	0.16666666666666663	0.006493506493506551
1.0	0.33333333333333337	0.16666666666666663	0.006493506493506551
1.0	0.6666666666666666	0.5833333333333334	0.08603896103896091
1.0	0.6666666666666666	0.5833333333333334	0.0
1.0	0.6666666666666666	0.5833333333333334	0.0
1.0	0.6666666666666666	0.5833333333333334	0.0
0.5	0.6666666666666666	0.5833333333333334	0.02922077922077926

Figura 21: Valores *contract_hours*, *contract_day*, *salary_real* y *type_retribution* escalados y normalizados

Una vez normalizadas las entradas en el rango de [0,1], tomando los valores numericos de los archivos *classification.csv* y *classificationemployee.csv*, se procede a calcular la entropía para descartar así correlaciones y redundancias, hacer que el modelo responda eficientemente (con certeza y velocidad) y decidir así los vectores que influirán en la clasificación y representación gráfica que realiza SOMPY, como se muestra en la sección Reducción de la dimensionalidad.

El cálculo de la entropía para los valores del empleado (Ver Figura 22) nos indica que las cuatro entradas analizadas: *contract_hours*, *contract_day*, *salary_real* y *type_retribution* son muy variables. A priori no se rechazó ninguna, ya que aportan valor al modelo.

```
Value entropy  contract_hours: 0.9998262077712042
Value entropy  contract_day: 0.8512563332302776
Value entropy  salary_real: 1.251912888274868
Value entropy  type_retribution: 0.676844964804416
unorganized values  [0.9998262077712042, 0.8512563332302776, 1.251912888274868, 0.676844964804416]
organized values  [0.676844964804416, 0.8512563332302776, 0.9998262077712042, 1.251912888274868]
final values  [0.9998262077712042, 0.8512563332302776, 1.251912888274868, 0.676844964804416]
columns to keep  [' type_retribution', ' contract_day', ' contract_hours', ' salary_real']
```

Figura 22: Resultados del cálculo de la entropía para los valores *contract_hours*, *contract_day*, *salary_real* y *type_retribution*

A partir de una distribución aleatoria de pesos y realizado el entrenamiento (Ver sección Aprendizaje), el algoritmo SOM evalúa el modelo y clasifica en función del mismo para así llegar a un mapa de zonas estables.

Con el objetivo de obtener una buena clasificación, los parámetros de la arquitectura de la red se configuraron con los siguientes valores:

- **data** (Colección de datos): Datos a agrupar, representados como una matriz de 252 filas x 5 columnas
- ***mapsize*** (dimensión del SOM): Matriz de 50x50
- ***normalization*** (normalización): var (Variable)
- ***initialization*** (inicialización del SOM): random (Aleatorio)
- ***neighborhood*** (vecindad): gaussian (Distribución gaussiana)
- ***mapshape*** (forma del SOM): planar
- ***mask***: None
- ***lattice*** (tipo de rejilla): rect (Rectangular)
- ***training*** (modo de entrenamiento): batch
- ***name*** (nombre usado para identificar el som): sompy

Durante el entrenamiento para el conjunto de entradas *Contract hours*, *Contract days*, *Salary* y *Type retribution*, se estableció un radius_ini de 4,166667, un radius_final de 1,000000 y un *trainlen* igual a 499 (Ver Figura 23), para una probabilidad de error de 0.000001 (Ver Figura 24).

```
Finetune training...
radius_ini: 4.166667 , radius_final: 1.000000, trainlen: 499

epoch: 1 ---> elapsed time:  0.246000, quantization error: 0.006048

epoch: 2 ---> elapsed time:  0.248000, quantization error: 0.050115

epoch: 3 ---> elapsed time:  0.252000, quantization error: 0.046985

epoch: 4 ---> elapsed time:  0.252000, quantization error: 0.046073

epoch: 5 ---> elapsed time:  0.250000, quantization error: 0.046088
```

Figura 23: Número de iteraciones, tiempo transcurrido y error de cuantificación, al inicio del entrenamiento

En ambos entrenamientos, tanto el descrito en la Figura 15 como el de la Figura 24 se observa que la probabilidad de error está por debajo de 0,02 que es el valor límite para inferir que el entrenamiento ha sido exitoso y por ende la clasificación también.

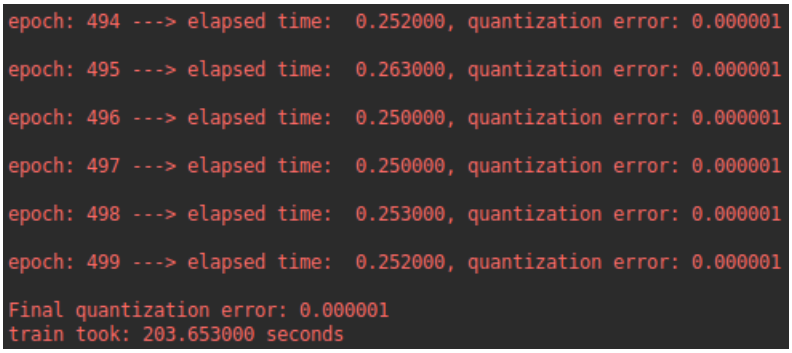


Figura 24: Número de iteraciones, tiempo transcurrido y error de cuantificación finalizado el entrenamiento

Finalmente, con los mapas resultantes de SOMPY que se muestran en la sección Evaluación y Clasificación se pretende descubrir las relaciones entre la variable *negative* anteriormente clasificada (Ver Figura 16) y el resto de variables contextuales, en este caso: *contract_hours*, *contract_day*, *salary_real* y *type_retribution*.

SOMPY asigna los colores en los mapas resultantes de manera aleatoria por cada una de las variables de entrada. Si la distancia promedio entre las entradas es alta, entonces los pesos circundantes son muy diferentes y se asigna un color claro a la ubicación del peso. Si la distancia promedio es baja, se asigna un color más oscuro.

En la Figura 25 se puede apreciar que las neuronas de salida en el caso de la variable *Contract hours* existe un gran número de empleados que se mueven en el mismo rango de horas (zonas oscuras). En el caso del mapa *Contract days* existe mayor variedad pero teniendo en cuenta que el *Grupo Faster* es una empresa de trabajo temporal, puede que esto sea un elemento determinante y uno de los factores que influyen en el sentimiento negativo de nuestros trabajadores.

Se puede observar además en el mapa para la variable *Contract days* que las zonas oscuras se reparten en la parte superior izquierda e inferior. Lo cual parece indicar que las zonas claras se diferencian perfectamente de los otros dos tipos.

En el mapa resultante de *Salary*, se aprecia que mayoritariamente las personas se mueve en el mismo rango salarial (zonas claras); pero esta información por sí sola no nos aporta nada, ya que por la naturaleza del modelo de negocio de Faster, el cual se basa en poner a disposición del cliente trabajadores nuestros, los salarios de estos trabajadores suelen estar muy ajustados a convenios teniendo pocas desviaciones. Lo cual cuadraría con la imagen del mapa resultante de la variable *Salary*, que estaría mostrando la uniformidad de este hecho; a priori dicho mapa aporta información útil, pero no estaría mal hacer alguna tarea de ML como *clustering* para encontrar comportamientos comunes entre las variables de entrada que apoyen al análisis y poder llegar a conclusiones más enriquecedoras y productivas.

Sin embargo es mucho más relevante el mapa *Type retribution* donde se observa una mayor concentración en las zonas oscuras, lo cual indica que existe un valor predominante, y al juzgar por el modelo de negocio bien podría ser retribución por horas o días; que son dos factores intrínsecamente relacionados con la volatilidad

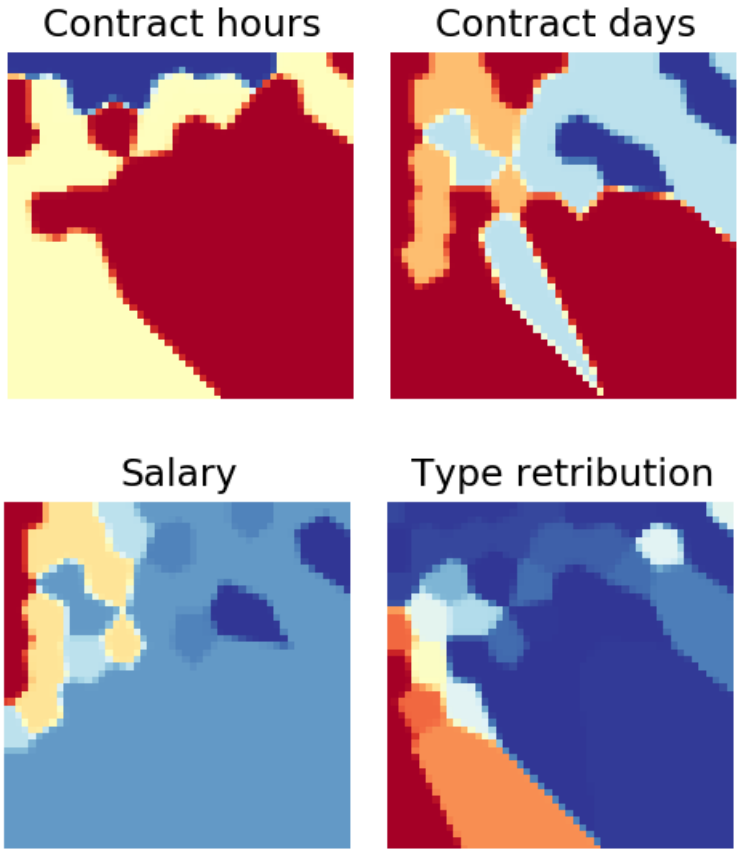


Figura 25: Mapas SOMPY para las entradas *contract_hours*, *contract_day*, *salary_real* y *type_retribution*

del contrato, que ejerce una presión negativa sobre la estabilidad profesional de una persona, y por tanto más relevante a la hora de asociar con ese de ánimo negativo detectado en el análisis del sentimiento.

Una característica de los mapas resultantes y en especial el mapa *Type retribution*, que es donde más se aprecia; es que cada color puede aparecer con dos posibles intensidades (zonas con el mismo color, unas más claras que otras) según la frecuencia de entradas asociados a una determinada neurona de salida.

Prototipo de la herramienta

En esta sección se introduce el prototipo de visualización, se detallan las estrategia de navegación y los enfoques interactivos planteados en la herramienta.

Contexto de la herramienta

El prototipo de la herramienta desarrollada propone analizar y visualizar los datos de los seguidores de *@FasterEmpleo* con el objetivo de entender el sentimiento de estos en *Twitter* y la causa o motivo del mismo. Se ha utilizado el enfoque propuesto en este trabajo, para identificar el sentimiento: positivo, negativo o neutral expresado por los tweets publicados.

En ese contexto, se han analizado los datos relacionados con los usuarios identicados como empleados del *Grupo Faster* con opiniones negativas. Dichos datos inicialmente estaban en la base datos del (CRM) Odoo [55]. Para ello se hizo necesario una tarea manual y así añadir dichos valores a la base datos de Twitter.

La implementación en su versión *beta* está soportada sobre un servidor dedicado con una distribución de Linux *Ubuntu*18,04, cuenta con 32*Gb* de RAM y 8 cores. En dicho servidor se encuentran además las bases de datos de PostgreSQL del CRM Odoo y la de RRSS Faster (Twitter).

Para la interfaz gráfica se optó por crear un Single Page Application (SPA) sobre Angular[56] ya que estas apps liberan al servidor de una parte del trabajo, reducen la cantidad de llamadas y mejoran la percepción de velocidad del usuario.

El intercambio de información entre la app Angular y los datos fluye a través de un servicio Representational State Transfer (REST), por ello se creó una Application Programming Interface (API), para establecer así la comunicación entre el *backend* (base de datos) y el *frontend* (SPA). Dicha API está desarrollada sobre el "micro" *Framework* Flask[57].

El objetivo de la visualización en esta herramienta es facilitarle al usuario final de forma intuitiva, en una vista general los datos estadísticos correspondientes a los tweets recopilados y en una vista más específica, donde se expresan más en detalle aspectos correspondientes con aquellos empleados cuyo estado de ánimo es negativo.

La herramienta está concebida como un sistema de apoyo a la toma de decisiones. Se articula en dos grandes apartados. Por un lado, se integran la información almacenadas en la base de datos referente a los *tweets* y los personas relacionadas con Faster, sobre el cual se pueden consultar los sentimientos, texto y localización. Y por otro lado, los datos estadísticos, la visualición gráfica y los mapas SOMPY con los resultados obtenidos de la investigación. Inicialmente es una implementación meramente informativa e interactiva, pero con vista futura a adicionar otras funcio-

[55] *https://odoo10.faster.es/*
[56] *https://angular.io/*
[57] *http://flask.pocoo.org/*

nalidades. Estas serían el registro de usuario vinculando las cuentas de usuarios del CRM Odoo, filtrar por localización lo cual es de gran utilidad ya que *Faster* cuenta con más de 30 delegaciones por toda España, etc.

La herramienta está disponible en la dirección localhost:8070 donde se puede trabajar e interactuar con la misma.

Estrategia de navegación

La estrategia de navegación empleada consiste en dos formas de visualización de la información: una global y otra más específica.

En la vista global se plantea un *dashboard*, que agrupa en una única vista datos relacionados con los *tweets*, y los usuarios almacenados en la base de datos, así como estadísticas de carácter general como cantidad de tweet positivos, negativos, neutrales y sus relativos valores porcentuales (Ver Figura 26).

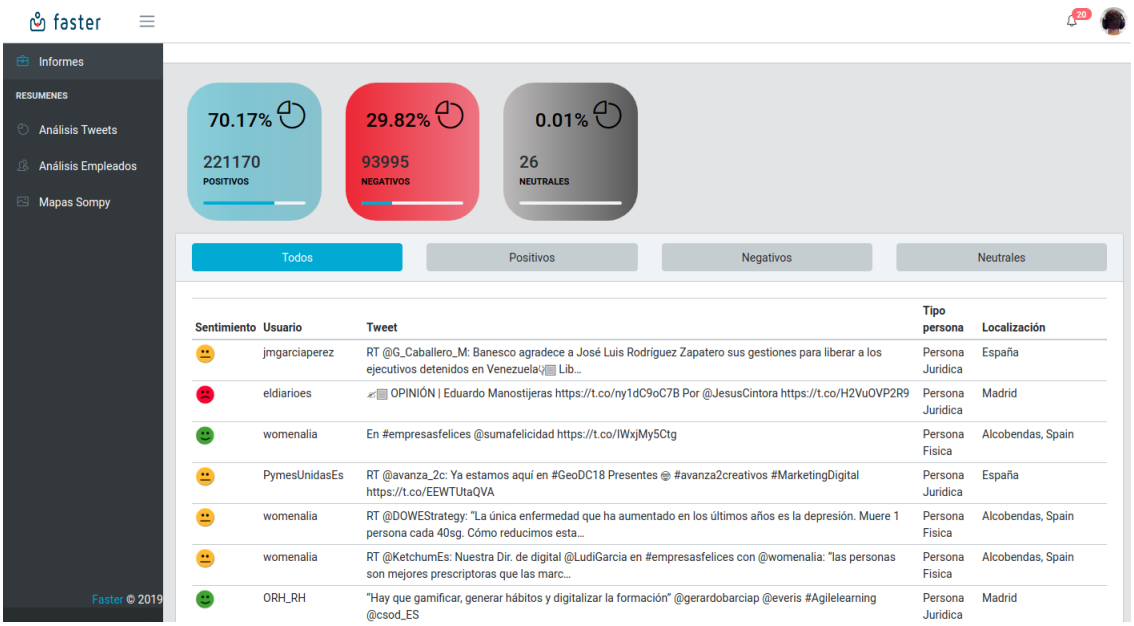


Figura 26: Visualización de vista global de la herramienta

En la parte superior derecha en el ícono de notificaciones se visualiza la cantidad de *tweets* nuevos a clasificar.

En la tabla que muestra el listado de *tweets* se visualizan el conjunto de ellos identificados según el *sticker* que refleja el sentimiento de cada mensaje (Ver Figura 27).

Figura 27: Visualización de mensajes filtrados por sentimiento positivo

Enfoque interactivo

El enfoque interactivo planteado en la herramienta permite la manipulación directa del analista con varios elementos de la visualización: como la tabla y los botones.

La *tabla* mostrada en la Figura 27, cuyo contenido son los tweets de la base de datos, donde se observan 5 atributos: *Sentimiento*, *Usuario*, *Tweet*, *Tipo*, *Localización*.

- **Sentimiento**: Muestra el sentimiento expresado en *stickers*, cada color representa un sentimiento: verde (positivo), rojo (negativo), amarillo (neutral)
- **Usuario**: Muestra el nombre del usuario
- **Tweet**: Muestra el mensaje publicado
- **Tipo Persona**: Muestra si la persona relacionado con el usuario es un cliente (Persona Jurídica) o un empleado (Persona Física).
- **Localización**: Señala la localización informada por el usuario.

Otra forma de interacción del usuario final con la herramienta es a través de los *botones* dispuesto por encima de la tabla, ilustrados en la Figura 27. Cada uno tiene una funcionalidad específica.

Los botones *Positivos*, *Negativos* y *Neutrales* actualizan la tabla, filtrando los mensajes por dichos sentimientos y visualizando los mismo en la tabla según la selección del usuario. El botón *Todos* restaura los cambios aplicados por los otros tres botones, actualizando el listado a su visualización inicial.

La herramienta también proporciona la visualización de estadísticas de los datos recogidos.

Los datos estadísticos relacionados con los *tweets* se pueden visualizar a través de la opción de menú *Análisis Tweets*, localizada en la parte lateral izquierda de la pantalla, como se muestra en la Figura 28.

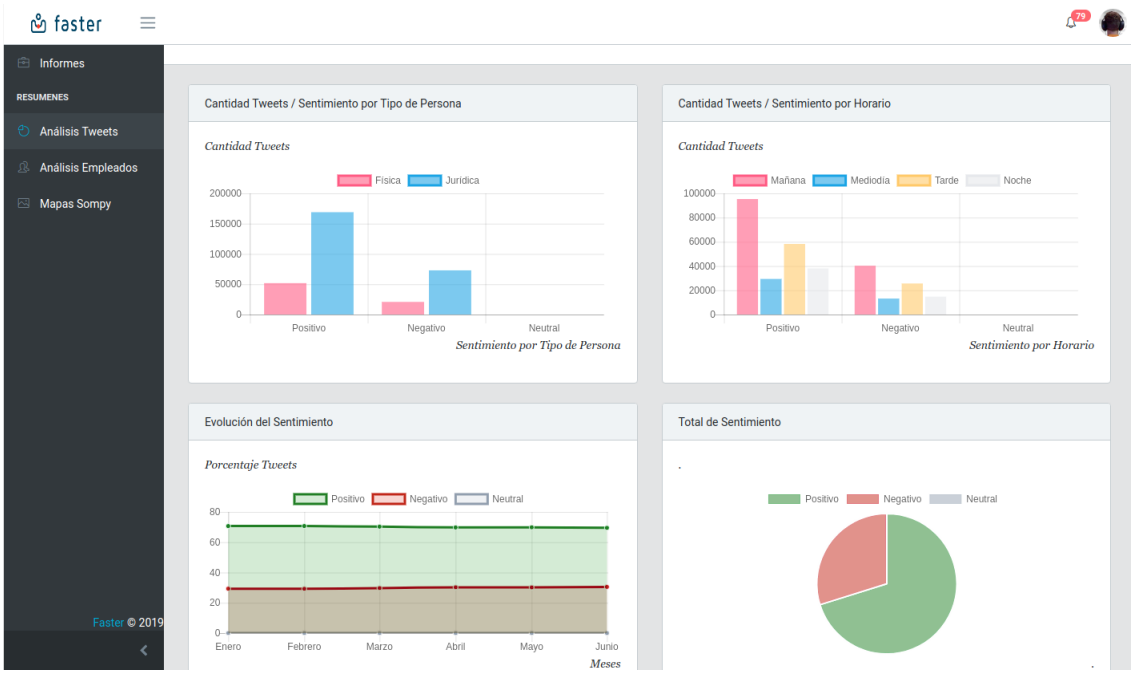


Figura 28: Visualización de la opción de menú: *Análisis Tweets*

En la Figura 28 se muestra una combinación de cuatro gráficos en forma de barras y de tarta o sectores, relacionados con: el *Sentimiento por tipo de persona*, *Sentimiento por horario*, *Evolución del sentimiento* en los últimos seis meses y *Total de Sentimiento*.

- **Sentimiento por tipo de persona**: Muestra el total de *tweet* clasificados en positivo, negativo y neutral según el tipo de persona.
- **Sentimiento por horario**: Muestra el total de *tweet* clasificados en positivo, negativo y neutral segúun el horario de publicación (mañana, mediodía, tarde, noche).
- **Evolución del sentimiento**: Muestra una gráfica de tiempo, con la evolución del sentimiento positivo, negativo y neutral expresado en los porcentajes en los últimos seis meses.
- **Total de Sentimiento**: Muestra una gráfica en forma de tarta agrupando los *tweets* por sentimiento.

Para la representación de la gráfica *Sentimiento por tipo de persona* se tuvo en cuenta la cantidad de *tweets* según el tipo de persona (coordenada de las Y) y los sentimientos clasificados en positivos, negativos y neutrales (coordenadas de las X).

Para la representación de la gráfica *Sentimiento por horario* se tuvo en cuenta la cantidad de *tweets* según el hora de publicación (coordenada de las Y) y los sentimientos clasificados en positivos, negativos y neutrales (coordenadas de las X).

Para la representación de la gráfica *Evolución del sentimiento* se tuvo en cuenta el porcentaje de *tweets* positivos, negativos y neutrales para cada uno de los meses, desde enero hasta junio (coordenada de las Y) y los meses a analizar (coordenadas de las X).

En los datos estadísticos de la Figura 28 se pueden observar que en las gráficas *Sentimientos por Tipo de Persona* y *Evolución del Sentimiento*, la sumatoria y porcentaje de los mensajes positivos lo cual representa el 70,17 %, son más de la mitad que la sumatoria de negativos (29,82 %) y neutrales (0,01 %). En la gráfica *Sentimiento por horario* se visualiza el horario pico donde los usuarios son más activos en Twitter así como el sentimiento de los mensajes. En el gráfico de tarta *Total de Sentimiento* podemos contrastar que efectivamente predomina las opiniones positivas por encima de las negativas, y que el sentimiento neutral no es muy común.

Los datos estadísticos relacionados con los usuarios identificados como empleados de *Faster* se pueden visualizar a través de la opción de menú *Análisis Empleados*, como se muestra en la Figura 29.

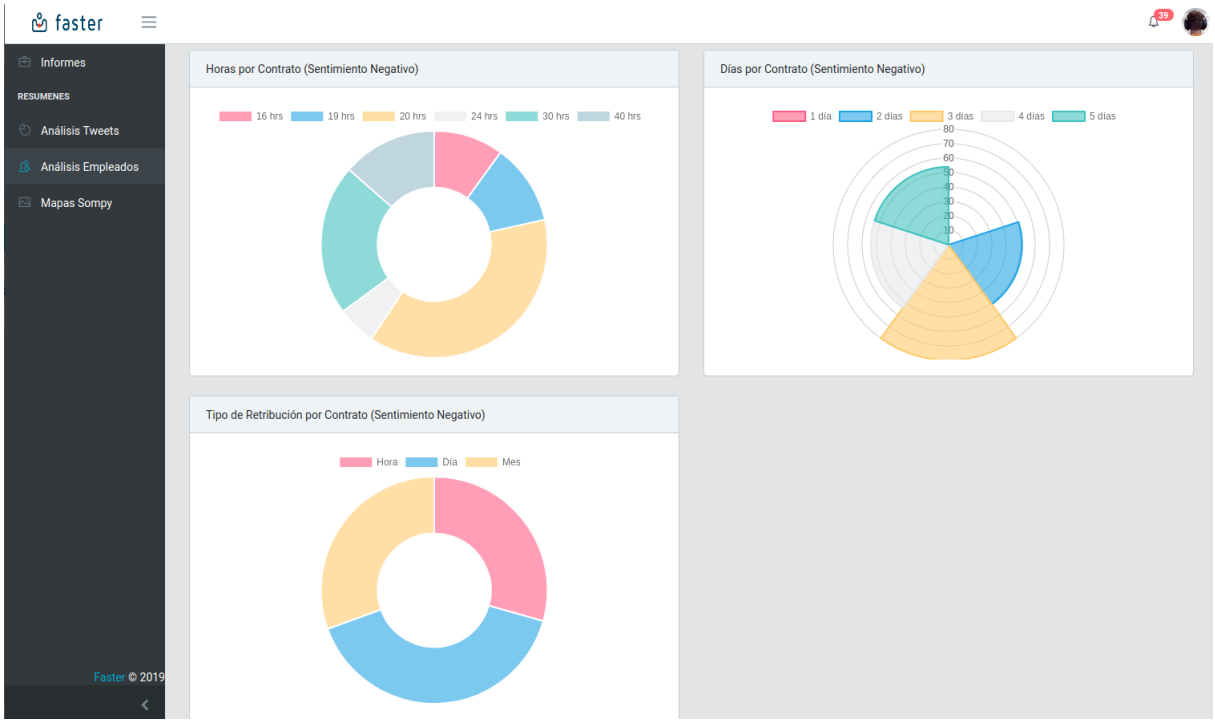


Figura 29: Visualización de la opción de menú: *Análisis Empleados*

En la Figura 29 se muestra tres gráficos en forma de tarta relacionados con los empleados de *Faster* con estado de ánimo negativo, y que se necesitan analizar: *Horas por Contrato*, *Días por Contrato* y *Tipo de Retribución por Contrato*. Dichas representaciones gráficas coinciden con los mapas resultantes SOMPY de la Figura 25.

Finalmente y como apoyo al sistema se muestra una opción de menú *Mapas SOMPY* donde se observan los mapas resultantes obtenidos durante la clasificación de los mensajes de Twitter, y los usuarios identificados como empleados del *Grupo*

Faster relacionados con dichos *tweets*, que se ha explicado en la sección Implementación del sistema.

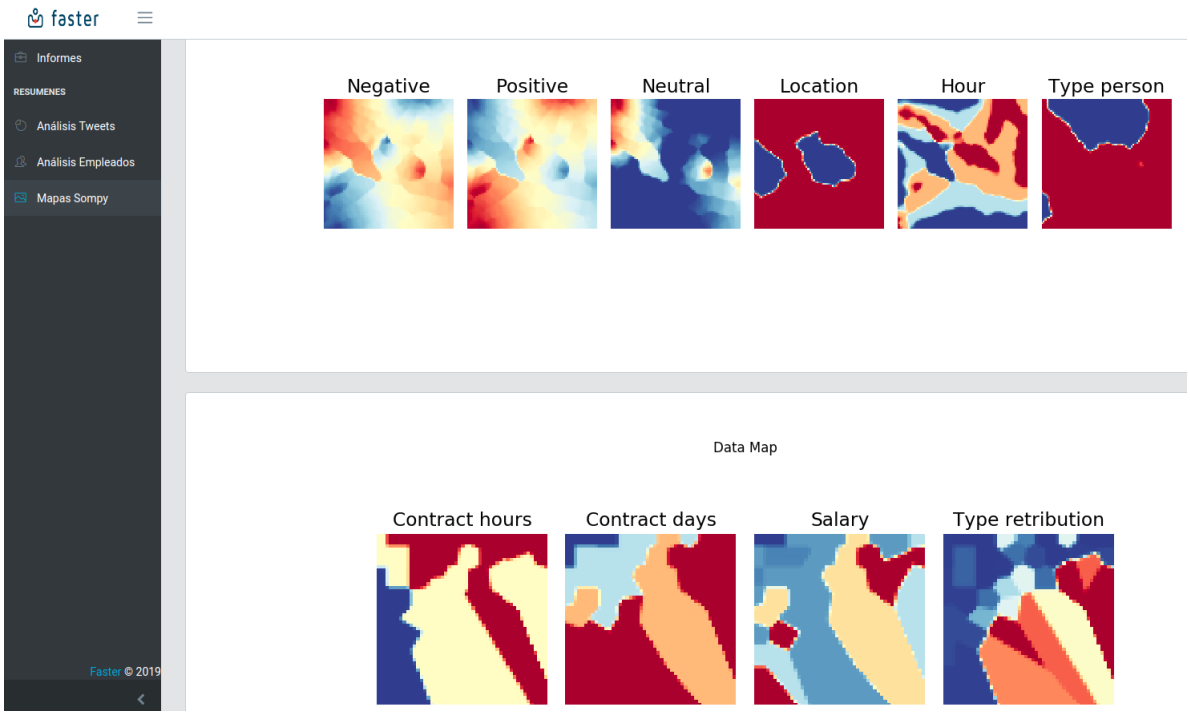


Figura 30: Visualización de la opción de menú: *Mapas SOMPY*

Conclusiones y Líneas Futuras de Investigación

En esta sección se sintetizan las conclusiones sobre el trabajo y las líneas futuras de investigación.

Conclusiones

En este trabajo se ha aplicado un nuevo enfoque al análisis de sentimiento. La idea subyacente es el uso de técnicas de minería de datos, algoritmos de ML para métodos no supervisados y el uso de Mapas Auto-organizados SOMPY.

La investigación ha permitido profundizar y comparar sistemas de minería de opiniones basados en redes sociales, a partir de técnicas de procesamiento del lenguaje natural y redes neuronales. Alcanzándose el objetivo propuesto de identificar y extraer de forma automática, información subjetiva como opiniones, preferencias, sentimientos y emociones de los usuarios. Para almacenarla de forma estructurada, poder procesarla y clasificarla como información útil. En este sentido hemos constatado la importancia de construir una base datos bien estructurada, con el fin de tener un conjunto de datos robusto para crear el modelo. El método en la recolección de datos ha sido muy efectivo, puesto que sería muy difícil extraer suficientes datos manualmente.

Se ha verificado que la tarea de procesamiento es esencial para el buen rendimiento del sistema, ya que elimina todos los términos que no aportan valor a la clasificación y mejora la precisión de los resultados.

Otra ventaja del método implementado es que mientras que en los métodos supervisados a la red hay que asignarle un conjunto de datos etiquetados, incluso manualmente, que lo ayuda a clasificar los datos en estas categorías. Con la ayuda del SOM se recibe el conjunto de datos y este aprende a clasificar sus componentes por sí mismo, evitando el esfuerzo manual.

Finalmente, todo ese proceso se plasma en una herramienta de visualización del prototipo propuesto, el cual ofrece una interfaz intuitiva, de fácil uso y enfocado en asistir al usuario final en la toma de decisiones. Permitiéndole encontrar patrones, mediante el uso de las gŕaficas estadísticas de tweets y empleados y acceder a esos datos.

A partir del Sistema de clasificación realizado y del análisis de los mapas SOMPY se pueden comenzar a tomar acciones comerciales con vista a mejorar el *Grupo Faster*.

Una vez detectada la fuente de la posible desmotivación laboral se trabajará en ello analizando posibles mejoras o realizando acciones correctivas que fomenten el bienestar entre los empleados, ya sea a través de formación que aumente sus conocimientos y seguridad, o bien a través de mejoras laborales en cuanto a condiciones laborales (horario, salario, cambio de centro, etc.).

Con los resultados obtenidos en la visualización gráfica, el Equipo Comercial y Dirección de *Faster* tratará de buscar mejoras en cuanto a la situación contractual de los trabajadores mediante el aumento de horas laborales para los casos de jornadas parciales o contratos por días, así como mejoras salariales, logrando una mayor estabilidad en su situación laboral que influya positivamente en su estado personal aumentando su bienestar y por lo tanto rebajando sus sentimientos u opiniones negativas.

En caso de no poder mejorar sus condiciones laborales marcadas por un convenio colectivo se realizarán acciones que ayuden a mejorar su sentimiento de pertenencia a la empresa ya sea a través de incentivos extras como tarjetas regalo en determinados negocios (Amazon, tiendas de ropa o alimentación), experiencias personales para su tiempo libre (smartbox, entradas de cine, etc); igualmente a través de comunicaciones internas felicitando su trabajo a toda la empresa; o bien a través de la organización de eventos sociales o reuniones de ocio de personal en los que hacerles partícipes.

Igualmente se pueden emprender acciones de evaluación de desempeño mediante las cuales el usuario valora aspectos de su trabajo tales como aptitudes y actitudes, tras lo cual será su responsable directo quien le evalúe en los mismos aspectos, pudiendo así recibir una retroalimentación de su trabajo en búsqueda de una mayor motivación.

Líneas Futuras de Investigación

Como líneas futuras de investigación, se propone estudiar soluciones para modelos supervisados.

Algo muy relevante a tener en cuenta es que una vez se obtenga el modelo funcional, se debería aumentar considerablemente el número de variables extraídas de la base de datos del CRM Odoo [58]. Y añadir registros como (trabajo, historial, tipo de contrato, cliente, etc). Ya que hasta ahora solo se ha trabajado con una matriz de muchas filas (tweets) pero pocas columnas (sólo las variables inferidas por la plataforma Twitter, y algunas otras relacionadas con el empleado).

[58] *https://odoo10.faster.es/*

Referencias

[1] **BARTOLOMÉ, A.**, *E-Learning 2.0-Posibilidades de la Web 2.0 en la Educación Superior*. Consultado el **14 de marzo de 2019**, en: http://www.lmi.ub.es/cursos/web20/2008upv [*Citado en pág. 1.*]

[2] **Malheiros, Y.** y **Lima, G.**, *Uma Ferramenta para Análise de Sentimentos em Redes Sociais Utilizando o SenticNet*. Simpósio Brasileiro de Sistemas de Informação, IX, **p. 517-522, 2013** [*Citado en pág. 1.*]

[3] **Fuji, Ren**, *From Cloud Computing to Language Engineering, Affective Computing and Advanced Intelligence*. University of Tokushima, Tokushima, Japan. International Journal of Advanced Intelligence, **Vol.2, N.1, p.1-14, July, 2010** [*Citado en pág. 2.*]

[4] **Vallez, M.** y **Pedraza-Jimenez, R.**, *El Procesamiento del Lenguaje Natural en la Recuperación de Información Textual y áreas afines*. Hipertext.net, núm. 5, 2007. Consultado el **15 de abril 2018** en: https://www.upf.edu/hipertextnet/numero-5/pln.html [*Citado en pág. 4.*]

[5] **Hutchins, W.J.**, *The Georgetown-IBM experiment demonstrated in January 1954*. En: Machine translation: from real users to research. 6th Conference of the Association for Machine Translation in the Americas, AMTA 2004, Washington, DC, USA, **September 28-October 2, 2004**; Editado por Robert E.Frederking y Kathryn B.Taylor. Berlin: Springer, 102-114. [*Citado en pág. 5.*]

[6] **Alcázar Jaén, S.**, *Diseño e implementación de un sistema para el análisis y categorización en Twitter mediante técnicas de clasificación automática de textos*. Universidad Carlos III de Madrid, **2013** [*Citado en pág. 5.*]

[7] **Pang, B.** y **Lee, L.**, *Opinion mining and sentiment analysis*. Foundations and Trends in Information Retrieval, p. 1-135 **2008** [*Citado en pág. 9.*]

[8] **Carr, N.**, *The Shallows. What the Internet is Doing to Our Brains*. W.W. Norton, **2010** [*Citado en pág. 6.*]

[9] **Terry Winograd**, *Procedures as a Representation for Data in a Computer Program for Understanding Natural Language*. MIT AI Technical Report 235, **February 1971**; Publicado: Cognitive Psychology Vol. 3 No 1, **1972** [*Citado en pág. 6.*]

[10] **Jones, K. S.**, *Natural Language Processing: a Historical Review*. Artificial Intelligence Review (2001), **Oct. 2001** [*Citado en pág. 6.*]

[11] **Brown, P.F., Cocke, J., Della Pietra, S.A., Della Pietra, V.J., Jelinek, F., Lafferty, J.D., Mercer, R.L.** y **Roossin, P.S.**, *A Statistical Approach to Machine Translation*. Computational Linguistics Volume 16, Number 2, **June 1990** [*Citado en pág. 6.*]

[12] **Otero, P.G.** y **González, M.G.**, *Técnicas de Procesamiento del Lenguaje Natural en la Recuperación de Información*. Centro de Investigación sobre Tecnoloxías da Lingua (CITIUS). Universidade de Santiago de Compostela, **2012** [*Citado en pág. 7.*]

[13] **Zhang, L., Ghosh, R., Dekhil, M., Hsu, M.** y **Liu, B.**, *Combining Lexiconbased and Learning-based Methods for Twitter Sentiment Analysis*. Hewlett-Packard Laboratories. Technical Report HPL-2011-89, **2011** [*Citado en págs. 9 y 23.*]

[14] **Pang, B., Lee, L.** y **Vaithyanathan, S.**, *Thumbs up? Sentiment classification using machine learning techniques*, en: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), p. 79-86, **2002** [*Citado en págs. 9 y 23.*]

[15] **Karinthy, F.**, *Chain-Links*, 1929. Consultado el **22 de abril de 2018**, en: https://www.goodreads.com/book/show/23491355-chains [*Citado en pág. 11.*]

[16] **O'Reilly, T.**, *What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software*, O'Reilly Media, Inc., **2005** [*Citado en pág. 10.*]

[17] **Mancera Rueda, A.** y **Pano Alamán, A.**, *Nuevas dinámicas discursivas en la comunicación política en Twitter*. Círculo de Lingüística Aplicada a la Comunicación 56, 53-80, **2013** [*Citado en pág. 12.*]

[18] **Benítez, R., Escudero, G., Kanaan, S.** y **Masip Rodó, D.**, *Inteligencia Artificial Avanzada*. Universitat Oberta Catalunya, Editorial UOC, **Julio 2013** [*Citado en págs.* v, v, *2, 4, 6, 8, 13, 14, 16, 17, 18, 19 y 20.*]

[19] **MARR, B.**, *What Is The Difference Between Deep Learning, Machine Learning and AI?*. Consultado el **29 de abril de 2019**, en: https://www.forbes.com/sites/bernardmarr/2016/12/08/what-is-the-difference-between-deep-learning-machine-learning-and-ai/#292da7ff26cf [*Citado en pág. 15.*]

[20] **Izaurieta, F.** y **Saavedra, C.**, *Redes Neuronales Artificiales*. Departamento de Física, Universidad de Concepción, Chile, Consultado el **29 de abril de 2019** en: https://www.academia.edu/11400964/Redes_Neuronales_Artificiales [*Citado en págs. 15 y 20.*]

[21] **PALMER POL, A.** y **MONTAÑO MORENO, J.J.**, *¿Qué son las redes neuronales artificiales? Aplicaciones realizadas en el ámbito de las adicciones*. Departamento de Psicología. Universidad de las Islas Baleares, España, Consultado el **29 de abril de 2019** [*Citado en págs. 20 y 21.*]

[22] **Lardinois, F.**, *Google launches its AI-powered jobs search engine*. Tech Crunch, Consultado el **02 de mayo de 2019**, en: https://techcrunch.com/2017/06/20/google-launches-its-ai-powered-jobs-search-engine/ [*Citado en pág. 22.*]

[23] **Shankar, S.** y **Lin, I.**, *Applying Machine Learning to Product Categorization*. Department of Computer Science, Stanford University, **2011** [*Citado en pág. 22.*]

[24] **Tony Mullen** y **Nigel Collier**, *Sentiment Analysis using Support Vector Machines with Diverse Information Sources*. Publicado en EMNLP **2004** [*Citado en pág. 9.*]

[25] **Casey Whitelaw**, **Navendu Garg** y **Shlomo Engelson Argamon**, *Using appraisal groups for sentiment analysis*. Publicado en CIKM **2005**, Consultado el **01 de mayo de 2019** [*Citado en pág. 9.*]

[26] **Bhavani Dasari, D.** y **Dr. Venu Gopala Rao. K**, *Text Categorization and Machine Learning Methods: Current State of the Art*. G. Narayanamma Institute of Technology and Science, Hyderabad, AP, India, **2012** [*Citado en pág. 22.*]

[27] **Thorsten Joachims**, *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*. Universität Dortmund, Informatik LS8, Dortmund, Germany. Consultado el: **02 de mayo de 2019** en: https://www.cs.cornell.edu/people/tj/publications/joachims_98a.pdf [*Citado en pág. 22.*]

[28] **Tripathy, Abinash**, *Classification of Sentiment of Reviews using Supervised Machine Learning Techniques*. International Journal of Rough Sets and Data Analysis (IJRSDA), **octubre 2016**. Consultado el: **03 de mayo de 2019** en: https://www.researchgate.net/publication/309209114_Classification_of_Sentiment_of_Reviews_using_Supervised_Machine_Learning_Techniques [*Citado en pág. 23.*]

[29] **Prechelt, L.** *An empirical comparison of C, C++, Java, Perl, Python, Rexx, and Tcl. Technical Report 2000-5*. Fakultät für Informatik, Universität Karlsruhe, Germany, March 2000. Consultado el: **30 de abril de 2019** [*Citado en pág. 37.*]

[30] **Universia España** *¿Qué es y para qué sirve Phyton?* Universia España, Fundación Universia, España, 19 de julio 2017. Consultado el: **30 de abril de 2019** [*Citado en pág. 37.*]

[31] **Hamilton, N.** *¿The A-Z of Programming Languages: Python. Computerworld.* Agosto 2008. Consultado el: **30 de abril de 2019** [*Citado en pág. 37.*]

[32] **Pedregosa, F.**, **Varoquaux, G.**, **Gramfort, A.**, **Michel, V.**, **Thirion, B.**, **Grisel, O.**, **Blondel, M.**, **Prettenhofer, P.**, **Weiss, R.**, **Dubourg, V.**, **Vanderplas, J.**, **Passos, A.** y **Cournapeau, D.** *Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research 12: 2825-2830., 2011.* Consultado el: **3 de mayo de 2019** [*Citado en pág. 39.*]

[33] **Gonzalez, L.**, *Aprende todo sobre Inteligencia Artificial*. Noviembre, 2018. Consultado el **4 de mayo 2019** en: http://ligdigonzalez.com/libreria-scikit-learn-de-python/ [*Citado en pág. 39.*]

[34] **Moosavi, V.**, *A Self Organizing Map (SOM) Package in Python: (SOMPY)*. Febrero, 2014. Consultado el **4 de mayo 2019** en: https://vahidmoosavi.com/2014/02/18/a-self-organizing-map-som-package-in-python-sompy/ [*Citado en págs. 40 y 41.*]

[35] **Denzer, P.**, *PostgreSQL*. Octubre de 2002. Consultado el **4 de mayo 2019** [*Citado en pág. 41.*]

[36] **Gonzalez Gil, J.**, *¿Qué es PostgreSQL?*. Agosto de 2018. Consultado el **4 de mayo 2019** en: https://openwebinars.net/blog/que-es-postgresql/ [*Citado en pág. 41.*]

[37] **Blanco, N.**, *¿Qué es Angular?*. Octubre de 2018. Consultado el **4 de mayo 2019** en: https://openwebinars.net/blog/que-es-angular/ [*Citado en pág. 43.*]

[38] **Domingo Muñoz, J.**, *¿Qué es Flask?*. Noviembre de 2017. Consultado el **4 de mayo 2019** en: https://openwebinars.net/blog/que-es-flask/ [*Citado en pág. 42.*]

[39] **Petrou, T.**, *Pandas Cookbook. Recipes for Scientific Computing, Time Series Analysis and Data Visualization using Python*. Octubre de 2017. Consultado el **18 de mayo 2019** [*Citado en pág. 41.*]

[40] **Simon, H.**, *Self-organizing maps. Neural networks - A comprehensive foundation (2nd edición). Prentice-Hall. ISBN 0-13-908385-5*. 1999. Consultado el **18 de mayo 2019** [*Citado en pág. 32.*]

[41] **Kohonen, T.**, *SOM Toolbox*. 2005. Consultado el **18 de mayo 2019** en: http://www.cis.hut.fi/projects/somtoolbox/theory/somalgorithm.shtml [*Citado en pág. 30.*]

[42] **Kohonen, T.**, **Honkela, T.**, *Kohonen network. Scholarpedia*. 2011. Consultado el **18 de mayo 2019** [*Citado en pág. 32.*]

[43] **Ultsch, A.**, **Siemon, H. Peter**, *Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis*. Paris, France, July 9–13, 1990. Consultado el **18 de mayo 2019** en: https://www.uni-marburg.de/fb12/arbeitsgruppen/datenbionik/pdf/pubs/1990/ultschsiemon90.pdf [*Citado en pág. 36.*]

[44] **Abhinav Ralhan**, *Self Organizing Maps*. Febrero 2018. Consultado el **22 de mayo 2019** en: https://towardsdatascience.com/self-organizing-maps-ff5853a118d4 [*Citado en pág. 21.*]

[45] **Marín, Juan M.**, *Los mapas auto-organizados de Kohonen (SOM)*. Septiembre 2007. Consultado el **25 de mayo 2019** en: http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/DM/tema5dm.pdf [*Citado en págs.* v *y 31.*]

[46] **Gervilla García, G.**, **Jiménez López, R.**, **Montaño Moreno, J.**, **Sesé, A.**,*The methodology of data mining. An application to the alcohol consumption in teenagers.*. Scientific Figure on ResearchGate, 2009. Consultado el **1 de junio 2019** en: https://www.researchgate.net/figure/Figura-1-Funcionamiento-general-de-una-neurona-artificial-y-su-representacion-matematica_fig3_28268241 [*Citado en págs.* v *y 21.*]

[47] **García Serrano, A.**, *Selección de atributos relevantes usando la entropía de Shannon*. Septiembre 2018. Consultado el **1 de junio 2019** en: https://www.ellaberintodefalken.com/2018/09/seleccion-atributos-relevantes-entropia-shannon.html [*Citado en págs. 29 y 30.*]

[48] **Wikipedia**, *Mapa autoorganizado*. Junio de 2019. Consultado el **24 de junio 2019** en: https://es.wikipedia.org/wiki/Mapa_autoorganizado#/media/Archivo:Somtraining.svg [*Citado en págs.* v *y 32.*]