

DETECCIÓN TEMPRANA DE POSIBLES CASOS DE DEPRESIÓN EN
COLOMBIA MEDIANTE EL ANÁLISIS DE TENDENCIAS EN TWITTER

JULIAN ENRIQUE FERNÁNDEZ FORERO

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2020

DETECCIÓN TEMPRANA DE POSIBLES CASOS DE DEPRESIÓN EN
COLOMBIA MEDIANTE EL ANÁLISIS DE TENDENCIAS EN TWITTER

JULIAN ENRIQUE FERNÁNDEZ FORERO

TRABAJO DE GRADO PARA OPTAR AL TÍTULO DE INGENIERO DE
SISTEMAS E INFORMÁTICA

DIRECTOR
SERGIO FERNANDO CASTILLO CASTELBLANCO
PH D.

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA

2020

DEDICATORIA

A mis padres, Luz Ángela Forero Leal y Mario Enrique Fernández Sarmiento por darme apoyo en mi formación académica y acompañarme durante todo este proceso.

A toda mi familia, a mis hermanos Mario Andrés y Carlos Eduardo, mi abuela Carolina (Q.E.P.D) quien estaría muy orgullosa de mí, mis amigos de colegio por siempre estar ahí, a mi novia Silvia Fernanda por el respaldo y a todas las personas que me apoyaron durante la carrera.

A la Coral Universitaria UIS por ser parte fundamental de mi formación integral y por regalarme momentos inolvidables de mi paso por la universidad.

Finalmente a mi director de proyecto, Sergio Fernando Castillo, a quien no solamente admiro profesionalmente sino también por su calidad humana.

AGRADECIMIENTOS

A la Universidad Industrial de Santander y especialmente a los docentes de la Escuela de Ingeniería de Sistemas e Informática, por todo el aporte a mi formación profesional y personal.

A mi director de proyecto, Sergio Fernando Castillo, por toda la colaboración, paciencia, confianza y por guiarme hacia la consecución de mis objetivos.

CONTENIDO

	Pág.
INTRODUCCIÓN	12
ESTRUCTURA DEL PROYECTO	13
DESCRIPCIÓN DEL PROYECTO	14
1.1 PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA.....	14
1.2 OBJETIVOS.....	15
1.2.1 Objetivo General	15
1.2.2 Objetivos Específicos.....	15
1.3 ALCANCE	15
MARCO DE REFERENCIA Y HERRAMIENTAS.....	17
2.1 MARCO TEÓRICO	17
2.1.1 Depresión.....	17
2.1.2 Procesamiento del lenguaje natural	18
2.1.3 Análisis de sentimientos.....	20
2.2 ESTADO DEL ARTE.....	22
2.3 HERRAMIENTAS	24
2.3.1 Jupyter Notebook.....	24
2.3.2 Librerías Python.....	24
2.3.3 API Twitter	25
2.3.4 Tweepy	26
2.3.5 Senpy.....	26
2.4 METODOLOGÍA	27
3. METODOLOGÍA PARA LA CLASIFICACIÓN DE TEXTOS.....	29
4. CONSTRUCCIÓN DEL MODELO	32
4.1 RECOLECCIÓN Y PRE PROCESAMIENTO DE TWEETS.....	32
4.1.1 Parametrización y autenticación	33
4.1.2 Almacenamiento de tweets	34
4.1.3 Tokenización y limpieza de tweets.....	35
4.1.4 Estemizado	35

4.2 EXTRACCIÓN DE CARACTERÍSTICAS	37
4.2.1 Filtrado por palabra clave.....	37
4.2.2 Clasificación por síntoma	40
4.2.3 Análisis de sentimientos.....	41
4.2.4 Clasificación manual de los tweets	45
4.2.5 Vectorización de textos	45
4.3 SELECCIÓN Y ENTRENAMIENTO DE CLASIFICADORES.....	47
4.3.1 Clasificadores seleccionados.....	48
4.3.1.1 Random Forest Classifier (RaF):	48
4.3.1.2 Multinomial Naive Bayes (MNB)	49
4.3.2 Entrenamiento de clasificadores	50
4.4 EVALUACIÓN DE CLASIFICADORES Y RESULTADOS	51
4.4.1 Métricas	51
4.4.2 Resultados obtenidos en el entrenamiento del modelo	53
4.4.2.1 Síntomas encontrados	54
4.4.2.2 Análisis de métricas	55
4.4.2.3 Frecuencia de repetición de palabras	58
5. PREDICCIONES DEL MODELO ENTRENADO.....	61
5.1 RECOLECCIÓN Y PRE PROCESAMIENTO DE TWEETS.....	61
5.2 EXTRACCIÓN DE CARACTERÍSTICAS	62
5.3 APLICACIÓN DEL MODELO Y VISUALIZACIÓN DE RESULTADOS	62
6. DIVULGACIÓN Y CUMPLIMIENTO DE OBJETIVOS	65
6.1 PARTICIPACIÓN EVENTO ACADÉMICO.....	65
6.2 PROPUESTA DE INVESTIGACIÓN.....	66
6.3 CUMPLIMIENTO DE OBJETIVOS	66
7. CONCLUSIONES Y TRABAJO FUTURO	68
7.1 TRABAJO FUTURO	69
BIBLIOGRAFÍA.....	70

LISTA DE TABLAS

Tabla 1: Palabras clave relacionados con tweets depresivos.....	38
Tabla 2: Código de clasificación de síntomas.....	40
Tabla 3: Resultados obtenidos en entrenamiento de clasificadores.	54
Tabla 4: Evaluación de clasificadores.....	58
Tabla 5: Resultados obtenidos para la ciudad de Medellín.....	62
Tabla 6: Cumplimiento de objetivos.....	67

LISTA DE FIGURAS

Figura 1: Arquitectura Senpy API.....	26
Figura 2: Fases del desarrollo evolutivo.	27
Figura 3: Fase 1 - Construcción del modelo	29
Figura 4: Fase 2 - Predicciones del modelo.....	31
Figura 5: Etapas de la fase 1 construcción del modelo.....	32
Figura 6: Sección de código - Parametrización y autenticación	33
Figura 7: Sección de código - Almacenamiento de tweets	34
Figura 8: Sección de código - Módulo de tratamiento de textos	36
Figura 9: Esquema de trastorno depresivo.	37
Figura 10: Sección de código - Filtrado de tweets	39
Figura 11: Sección de código – Clasificación por temática.....	41
Figura 12: Ontología Marl.	42
Figura 13: Captura de pantalla ejemplo de asignación de polaridad	43
Figura 14: Sección de código - Análisis de sentimientos	44
Figura 15: Ejemplo de creación de bolsa de palabras	46
Figura 16: Sección de código - Vectorización de los textos.....	47
Figura 17: Ejemplo de clasificación a través de un árbol de decisión.....	49
Figura 18: Sección de código - Entrenamiento de los clasificadores	50
Figura 19: Sección de código – Métricas del modelo entrenado	53
Figura 20: Sección de código - Visualización de temáticas encontradas.....	54
Figura 21: Síntomas encontrados en el entrenamiento de clasificadores, siendo ‘estado depresivo’ el valor 1, ‘falta de energía’ el valor 2, ‘falta de energía y depresivo’ el valor 3, ‘insomnio’ el valor 4, ‘insomnio y depresivo’ el valor 5, ‘insomnio y falta de energía’ el valor 6 y ‘todas las anteriores’ el valor 7	55
Figura 22: Sección de código - Matriz de confusión	56
Figura 23: Matriz de confusión del clasificador Random Forest.	56
Figura 24: Matriz de confusión del clasificador Multinomial Naive Bayes.	57
Figura 25: Sección de código - Visualización de frecuencia de repetición.....	59

Figura 26: Frecuencia de repetición de palabras estemizadas con datos de entrenamiento.....	59
Figura 27: Etapas de la fase 2 predicciones del modelo.....	61
Figura 28: Sección de código - Casos identificados en la ciudad de Medellín.....	63
Figura 29: Síntomas encontrados en la ciudad de Medellín, siendo ‘estado depresivo’ el valor 1, ‘falta de energía’ el valor 2, ‘falta de energía y depresivo’ el valor 3, ‘insomnio’ el valor 4, ‘insomnio y depresivo’ el valor 5, ‘insomnio y falta de energía’ el valor 6 y ‘todas las anteriores’ el valor 7	64
Figura 30: Palabras estemizadas con mayor frecuencia en la ciudad de Medellín	64
Figura 31: Participación en evento académico.	65

RESUMEN

Título: Detección temprana de posibles casos de depresión en Colombia mediante el análisis de tendencias en Twitter*

Autor: Julian Enrique Fernández Forero**

Palabras clave: Inteligencia artificial, depresión, twitter.

Descripción: Considerada por la Organización Mundial de la Salud como uno de los trastornos más debilitantes para el ser humano, la depresión se ha convertido en un problema de salud pública por el impacto que tiene en el individuo, su familia y la comunidad. Los trastornos mentales y neurológicos representan el 22% de la carga total de enfermedades en América latina y el Caribe (OPS) y representan una elevada carga de la enfermedad en términos de morbilidad, mortalidad y discapacidad.

Por otra parte, las personas utilizan cada vez más las plataformas de redes sociales para compartir sus pensamientos, deseos más profundos y expresar su opinión sobre asuntos sociales. Las publicaciones en estos sitios se realizan de forma natural y, por lo tanto, ofrecen una solución a la manipulación que a menudo encuentran los cuestionarios de depresión autoinformados.

De esta manera las redes sociales proporcionan un medio para capturar el estado mental presente de un individuo, e incluso son efectivas para representar sentimientos de inutilidad, culpa, impotencia y los niveles de odio hacia uno mismo que a menudo caracterizan la depresión clínica. Así, las redes sociales pueden utilizarse para detectar e incluso predecir el trastorno depresivo mayor, y posiblemente incluso complementar y ampliar los enfoques tradicionales para el diagnóstico de depresión.

*Trabajo de grado

**Facultad Ingeniería Físico-Mecánicas. Escuela de Ingeniería de Sistemas. Director Ing. Sergio Fernando Castillo Castelblanco.

ABSTRACT

Title: Early detection of possible cases of depression in Colombia by analyzing trends on Twitter*

Author: Julian Enrique Fernández Forero**

Key words: Artificial intelligence, depression, twitter.

Description: Considered by the World Health Organization as one of the most debilitating disorders for humans, depression has become a public health problem due to the impact it has on the individual, his family and the community. Mental and neurological disorders represent 22% of the total disease burden in Latin America and the Caribbean (PAHO) and represent a high disease burden in terms of morbidity, mortality and disability.

On the other hand, people are increasingly using social media platforms to share their thoughts, deepest desires and express their opinion on social issues. Publications on these sites are made naturally and, therefore, provide a solution to the manipulation often found by self-reported depression questionnaires.

In this way social networks provide a means to capture the present mental state of an individual, and are even effective in representing feelings of worthlessness, guilt, helplessness and levels of self-hatred that often characterize clinical depression. Thus, social networks can be used to detect and even predict major depressive disorder, and possibly even complement and expand traditional approaches to the diagnosis of depression.

*Bachelor Thesis.

** Facultad Ingeniería Físico-Mecánicas. Escuela de Ingeniería de Sistemas. Director Ing. Sergio Fernando Castillo Castelblanco.

INTRODUCCIÓN

Las redes sociales en la actualidad son un mecanismo importante de socialización que permite la interacción entre individuos, el intercambio de ideas, pensamientos y demás información que en esta era digital es primordial para entender el mundo en el que vivimos. De esta manera, se convierten en un foco de información que alberga datos valiosos acerca de la identidad y comportamiento de sus usuarios y pueden ser usados para identificar factores de riesgo en temas de salud mental como la depresión.

Desde problemas alimenticios y adicción a las drogas hasta intentos de cometer suicidio, son los temas que evidencian que en ocasiones los individuos escogen las redes sociales para desahogarse antes de acudir a un profesional. La libertad que brindan plataformas como Twitter facilita que la información que allí se publica no esté condicionada, sino bien, que esta sea voluntariamente expuesta por cada uno de los usuarios que la utiliza. Así, el análisis de estos datos puede ser útil para identificar síntomas de trastornos como la depresión y de esta manera predecir y monitorear posibles futuros casos.

En consecuencia, haciendo uso de los datos disponibles en redes sociales junto con algoritmos de aprendizaje automático y procesamiento del lenguaje natural, el presente proyecto pretende identificar los síntomas más comunes de la depresión (falta de energía, estado de ánimo deprimido e insomnio)¹ en usuarios de Twitter en Colombia para predecir posibles casos futuros. Brindar alternativas en áreas de la salud para identificar tempranamente estos trastornos mentales en usuarios posibilitará una ayuda efectiva a la hora de tratarlos, además de facilitar un control estadístico de casos presentados para realizar seguimiento y proveer opciones de mecanismos de prevención.

¹ (Mowery et al, 2016)

ESTRUCTURA DEL PROYECTO

A continuación se presentan los diferentes capítulos en los cuales está planteado y desarrollado el proyecto.

Capítulo 1. Descripción del proyecto: Se encuentra el planteamiento y descripción del problema, así como los objetivos y alcance del proyecto.

Capítulo 2. Marco teórico y herramientas: Se presenta el marco de referencia, marco teórico, herramientas y metodología utilizada.

Capítulo 3. Metodología para la clasificación de textos: Se encuentran las fases en las cuales se llevará a cabo el proyecto, así como la descripción de los procesos realizados en cada una de ellas.

Capítulo 4. Construcción del modelo: En este capítulo se presenta la recolección y pre procesamiento de los tweets, la extracción de características, selección y entrenamiento de algoritmos y finalmente la evaluación de clasificadores y resultados.

Capítulo 5. Predicciones del modelo entrenado: Se encuentra todo lo relacionado con la aplicación del modelo en otro conjunto de datos.

Capítulo 6. Divulgación y cumplimiento de objetivos: Este capítulo presenta la participación en evento académico, la propuesta de investigación realizada y el cumplimiento de objetivos.

Capítulo 7. Conclusiones y trabajo futuro.

DESCRIPCIÓN DEL PROYECTO

1.1 PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA

De acuerdo con la Organización Mundial de la Salud (OMS), la salud mental y física y el bienestar social son componentes vitales intrínsecamente ligados a todo ser humano. No obstante, no se le ha atribuido a la salud mental la importancia necesaria para identificar trastornos neuropsiquiátricos que puedan afectarla como la depresión, un desorden mental que afecta a millones de personas en el mundo.

Comúnmente la detección de estos casos se deriva de experiencias reportadas por los mismos pacientes, cuestionarios de comportamiento, encuestas o una valoración de salud mental por parte de un profesional. En Colombia, el último boletín de salud mental de la subdirección de enfermedades no transmisibles del Ministerio de Salud, indica los estudios poblacionales realizados que incluyen información acerca de la depresión, como lo son la Encuesta Nacional de Salud Mental de 2015 (ENSM) y la encuesta SABE 2014-2015. Sin embargo, los resultados de estos exámenes están basados en las respuestas de los mismos usuarios y por ello pueden estar manipuladas para ocultar o maquillar información. La OMS reporta que la mayoría de individuos con casos de depresión no buscan un tratamiento, y que aún en consultas médicas esta puede pasar desapercibida y por ende no es diagnosticada a tiempo, siendo la causa número uno de suicidios hoy en día.

De esta manera, se buscan alternativas que proporcionen herramientas para identificar de manera temprana síntomas de posibles casos de depresión en Colombia, donde el Sistema Integral de Información de la Protección Social (SISPRO) reporta que las atenciones a causa de este trastorno se han incrementado desde el año 2009.

1.2 OBJETIVOS

1.2.1 Objetivo General. Desarrollar un prototipo software que permita, mediante técnicas de aprendizaje automático (machine learning), clasificar publicaciones provenientes de la red social Twitter para así identificar tendencias de depresión en Colombia.

1.2.2 Objetivos Específicos

- 1) Diseñar y programar las funcionalidades correspondientes al descargue y pre-procesamiento de los trinos.
- 2) Diseñar y programar un analizador de texto que identifique las temáticas y asigne la polaridad a los textos pre-procesados.
- 3) Seleccionar dos algoritmos de aprendizaje automático a utilizar en el prototipo, de los más usados en el área del procesamiento de lenguaje natural, a saber: Naive Bayes, Random Forest, Regresión Logística, Support Vector Machine.
- 4) Realizar la clasificación de los textos mediante los dos algoritmos seleccionados y evaluar los correspondientes resultados.
- 5) Diseñar y programar la visualización de resultados.

1.3 ALCANCE

Se desarrollará un prototipo que permita la recolección en tiempo real de tweets bajo ciertos parámetros, se creará un módulo que permita su pre procesamiento, procesamiento y finalmente se entrenarán algoritmos de aprendizaje automático

para detectar posibles casos relacionados con la depresión. El objetivo del presente proyecto es el estudio de los algoritmos entrenados, para evaluar su comportamiento mediante métricas que permitan determinar su eficacia a la hora de identificar posibles casos de depresión en Twitter.

El desarrollo de este prototipo envuelve un problema de clasificación de mensajes de usuarios de esta red social, por lo que se ha decidido evaluar sólo algoritmos supervisados, los cuales se entrenarán de manera manual, produciendo una correspondencia entre las características de clasificación proporcionadas y las deseadas. Tanto el desarrollo como los resultados obtenidos estarán disponibles a través de Jupyter Notebooks para ser consultados.

La depresión es una condición significativamente compleja, y el profesional que puede hablar con propiedad de dicha enfermedad es un médico con especialización en psiquiatría. En el presente trabajo se toma una visión ingenieril del problema y no se trabajó con expertos en psiquiatría, dado que no se contaba con dicho tipo de recursos. Así, la detección de depresión es la detección de palabras típicamente relacionadas con la depresión como “Llorar, triste, abandonado, soledad” y demás descritas en la sección 4.2.1.

De igual forma, se realiza un análisis sintáctico a nivel de palabras, no se trabaja ni el contexto ni la pragmática. Por ejemplo, *“leí una novela buenísima que recomiendo, se llama lágrimas en el mar”*, será tomado como si fuera una publicación depresiva (dado que se localiza la palabra *“lágrimas”* en el texto). Este tipo de mejoras se plantean para versiones futuras del presente trabajo y son presentadas en el capítulo 7, conclusiones y trabajo futuro.

MARCO DE REFERENCIA Y HERRAMIENTAS

Actualmente el análisis de datos a través de algoritmos de aprendizaje automático, y el procesamiento del lenguaje natural aplicado a datos provenientes de redes sociales se realiza en diversas áreas como el mercadeo, detección de casos de violencia y discriminación, insomnio, pensamientos suicidas, entre otros; esto con el fin de generar datos estadísticos, tener conocimiento de tendencias y en otros casos poder tomar medidas correctivas y preventivas como es el caso de la salud mental.

Cuando se accede a una red social se permite el acceso y uso de los datos que allí se publican, lo cual posibilita moderar el contenido, entregar información valiosa a las empresas o ser utilizada con fines académicos como es el caso.

2.1 MARCO TEÓRICO

2.1.1 Depresión. La depresión se define como un trastorno mental, que se caracteriza por la presencia de tristeza, pérdida de interés o placer, sentimientos de culpa o falta de autoestima, trastornos del sueño o del apetito, sensación de cansancio y falta de concentración². Esta se puede presentar de manera leve dificultando el diario vivir y desempeño de las personas que la padecen, y de manera más grave llevando a cometer suicidio si no se trata de la manera adecuada.

Esta condición mental puede darse tanto en hombres como en mujeres en cualquier edad y dependiendo de esto varían sus síntomas. Los niños pueden experimentar síntomas depresivos intermitentes demostrando episodios de irritabilidad,

² ORGANIZACIÓN MUNDIAL DE LA SALUD: Depresión [En línea]. [Consultado: 1 de agosto de 2019] Disponible en: <https://www.who.int/topics/depression/es/>

negatividad o estado de ánimo malhumorado, mientras que en los adultos mayores se puede presentar de manera más persistente, debido a enfermedades crónicas o etapas de duelo por la muerte de un ser querido y es más común que traten de ocultarlo haciendo más difícil el diagnóstico. Como se mencionó anteriormente en la sección 1.3, *ALCANCE DEL PROYECTO*, el presente trabajo adopta una visión ingenieril del problema haciendo un análisis sintáctico a nivel de palabras, sin contar con la participación de un profesional de la salud mental y sin tener en cuenta la pragmática de los tweets recolectados.

2.1.2 Procesamiento del lenguaje natural. La comunicación entre elementos de la misma naturaleza se da de manera sencilla y directa, siendo un elemento esencial a la hora de crear relaciones entre individuos. Dicho esto, y debido al afán de generar una relación entre humanos y computadores se hace necesaria la creación de protocolos como los lenguajes naturales que faciliten esta interacción, y es allí donde aparece el procesamiento del lenguaje natural.

El procesamiento del lenguaje natural es un campo de las ciencias de la computación, inteligencia artificial y lingüística que estudia las interacciones entre las computadoras y el lenguaje humano. Los modelos aplicados se enfocan en la comprensión del lenguaje, los aspectos cognitivos y la organización de la memoria, y de esta manera permite extraer características medibles de textos escritos por humanos e identificar reglas básicas y patrones establecidos en el lenguaje³.

En consecuencia es fácil deducir que es una disciplina que cuenta con un alto potencial y múltiples posibilidades prácticas, tantas como lenguajes naturales poseen. Su cometido es mejorar y hacer eficaz la comunicación entre personas y computadores. Por esta razón, cualquier área asociada al lenguaje y las relaciones entre humanos y máquinas se puede ver afectada positivamente por el

³ (SOBRINO José, 2018)

procesamiento del lenguaje natural. Aunque sus aplicaciones son innumerables, de acuerdo a Sobrino Sande, algunas de las más importantes son:

- **Recuperación de la información:** el objetivo de este tipo de sistemas es la búsqueda y obtención de grupos de documentos electrónicos a partir de un conjunto de palabras clave proporcionadas por el usuario. Los documentos devueltos normalmente se ordenan con base a algún tipo de atributo que mide su relevancia dentro del resultado global. Estos sistemas son en los que basan su funcionamiento los buscadores de internet y representan la primera aplicación implantada masivamente dentro del mundo de las tecnologías de la información.
- **Traducción automática de textos:** Los sistemas actuales de traducción automática utilizan un enfoque basado en mediciones estadísticas y relaciones entre textos a partir de un entrenamiento previo con cientos o miles de textos. Aunque las traducciones no son perfectas y no pueden sustituir a humanos en textos complejos en los que se requiera alta fidelidad, si son aceptables para tareas como traducir mensajes de una red social o una página web.
- **Reconocimiento del habla:** Este tipo de sistemas permiten a las personas interactuar con los ordenadores u otros dispositivos electrónicos como teléfonos inteligentes o automóviles mediante el uso de un lenguaje natural y por medio de la voz. Un ejemplo de ello son los llamados “asistentes virtuales” como Cortana, Siri, Google Assistant o los servicios de atención al cliente habituales en actividades bancarias y telecomunicaciones. Virtualmente todo dispositivo electrónico podría en el futuro poseer un sistema de reconocimiento del habla de manera que sus funciones pudiesen ser controladas mediante comandos de voz. Por tanto, el campo del reconocimiento de la voz humana se postula como uno de los más importantes debido a la popularización de este tipo de sistemas.

- **Extracción de la información:** Este tipo de tareas consiste en analizar textos o mensajes con el objetivo de capturar y extraer automáticamente aquella información considerada de interés. Estos documentos pueden ser anuncios, artículos de prensa, informes de carácter científico, etc., y los datos a extraer, nombres de personas, organizaciones, teléfonos, fechas u otros. El proceso de extracción de la información es básico para poder clasificar documentos, resumirlos o relacionarlos entre sí.
- **Análisis de sentimientos:** Ofrece la posibilidad de conocer automáticamente cuál es la opinión que una persona tiene sobre un determinado tema a partir de las ideas expresadas en un texto. Este sentimiento u opinión es una valoración cualitativa o cuantitativa acerca de un producto, servicio, persona o cualquier otro tipo de entidad⁴.

2.1.3 Análisis de sentimientos. Dentro del campo del procesamiento del lenguaje natural, el análisis de sentimientos busca mediante técnicas computacionales analizar información subjetiva expresada en un texto determinado, y de esta forma saber si un texto presenta connotaciones positivas o negativas. Sus tareas principales según Sobrino Sande son:

- **Extraer y categorizar entidades:** dado un documento, en primer lugar, se deben encontrar e identificar todas las entidades que contiene y agruparlas con base en su significado común.
- **Extraer y categorizar aspectos:** en esta tarea se buscarán y capturarán los aspectos del documento teniendo en cuenta que pueden existir distintas formas de expresarlos. Una vez localizados, se deben extraer y agrupar para, a continuación, asociar cada grupo con su entidad correspondiente.
- **Extraer el momento temporal:** se trata de detectar el momento en el que la opinión fue emitida.

⁴ Ibid, p. 21.

- **Clasificar la polaridad a nivel de aspecto:** para cada par de entidad y aspecto se debe determinar la valoración emitida por el autor de la opinión.⁵

Adicionalmente, Sobrino indica que este proceso se puede llevar a cabo a tres niveles distintos con base a su granularidad, profundidad y detalle requeridos. Estos niveles son:

- **Análisis a nivel de documento:** en este nivel se analiza el sentimiento global de un documento como un todo indivisible, clasificándolo como positivo, negativo o neutro o usando otro sistema de clasificación. En estos casos, se asume que dicho documento expresa una valoración sobre una única entidad, por lo que no es aplicable en aquellos que hablen sobre varias entidades simultáneamente.
- **Análisis a nivel de oración:** en este caso, se divide el documento en oraciones individuales para extraer posteriormente la opinión que contiene cada una de ellas. La opinión de cada oración puede ser, de nuevo, positiva, negativa o neutra o bien tomar un valor con base en otro tipo de medida.
- **Análisis a nivel de aspecto y entidad:** este es el nivel de análisis con mayor detalle posible, en donde una entidad está formada por distintos elementos o aspectos y sobre cada uno de ellos se expresa una opinión cuya polaridad puede ser distinta en cada caso.⁶

Finalmente Sobrino⁷ expresa que se debe tener en cuenta que además de la dificultad que implica ejecutar correcta y eficazmente las tareas anteriores para el análisis de sentimientos, esta área de investigación presenta una serie de problemas y obstáculos. La detección de figuras como el sarcasmo y la ironía, que en muchas ocasiones son difíciles hasta para los seres humanos, así como las

⁵ Ibid, p. 28

⁶ Ibid, p. 30

⁷ Ibid, p. 30

expresiones coloquiales y ambigüedades en una oración, y el correcto tratamiento de intensificadores y negaciones representa un reto para el análisis de sentimiento. Además, el contexto en el cual encuentra una palabra es también determinante a la hora de definir su polaridad, así, una misma palabra puede expresar sentimientos positivos en un contexto y negativos en otro.

2.2 ESTADO DEL ARTE

Con la creciente problemática en salud mental de trastornos como la depresión, el auge de las redes sociales y la evolución de herramientas como el procesamiento del lenguaje natural y los algoritmos de aprendizaje automático, estos mecanismos se pueden aprovechar para apoyar el análisis de conjuntos de datos muy grandes para la investigación de salud mental a nivel de población⁸. Por ejemplo, utilizando datos de redes sociales, los investigadores han caracterizado problemas de adicción a bebidas alcohólicas o cigarrillo⁹, fases clasificadas de adicción a sustancias¹⁰, predijeron la probabilidad de recuperarse de un trastorno alimentario¹¹, e identificaron individuos en riesgo de suicidarse¹². De esta manera se han realizado diferentes trabajos investigativos que apuntan hacia las estrategias para la identificación de diferentes casos en Twitter.

*“Towards automatically classifying depressive symptoms from Twitter data for population health”*¹³ es un estudio enfocado en la identificación de síntomas basados en un dataset de factores psicosociales y depresivos llamado *“Depressive Symptoms and Psychosocial Stressors Associated with Depression”* (SAD) con 9.300 tweets, del cual se extrajeron palabras clave indicadoras de depresión a través de un profesional de la salud. Cada uno de los tweets fue clasificado en una

⁸ (Conway y O'Connor, 2016)

⁹ (Tamersoy et al., 2015; Myslín et al., 2013)

¹⁰ (MacLean et al., 2015)

¹¹ (Chancellor et al., 2016)

¹² (De Choudhury et al., 2016)

¹³ (Mowery et al, 2016)

o más clases de acuerdo con el esquema de anotación lingüística del Manual Diagnóstico y Estadístico de Trastornos Mentales, entre los cuales se destacaron el estado de ánimo deprimido, dificultad para dormir y la fatiga o pérdida de energía.

Por otro lado, “*How well do Spaniards sleep? Analysis of Sleep Disorders based on Twitter mining*”¹⁴ estudio realizado por la Universidad Politécnica de Madrid, profundiza más acerca de la obtención de los datos a través de librerías como Tweepy y la API Streaming de Twitter condicionados por ciertas reglas, consiguiendo 54.432 tweets. Una vez filtrados por palabra clave redujeron el tamaño de sus datos enfocándose en una región en concreto gracias a Google Maps Geocoding API, extrayendo 5.112 trinos de usuarios españoles para procesar.

Ambos estudios, incluyen técnicas de procesamiento de lenguaje natural para procesar los tweets, así como algoritmos de aprendizaje automático para predecir posibles casos de trastornos mentales y evaluar su eficacia a la hora de realizar esta tarea. También es el caso del estudio “*Identifying Depression on Twitter*”¹⁵ de la Universidad de Johns Hopkins en conjunto con la Universidad de Tulsa, quienes a través de un dataset creado para la conferencia “*Computational Linguistic and Clinical Psychology*” analizaron el comportamiento de diferentes algoritmos de clasificación, concluyendo que la red social Twitter en conjunto con estos procesos y herramientas tienen un gran potencial para predecir casos de depresión.

Algunos de los algoritmos más utilizados en estos estudios fueron Random Forest, Support Vector Machine, Logistic Regression y Naive Bayes. Todos estos bajo distintas condiciones pero puestos a prueba con las mismas métricas que determinarían el éxito o fracaso de sus predicciones.

¹⁴ (Suárez et al, 2018)

¹⁵ (Nadeem et al, 2016)

2.3 HERRAMIENTAS

A continuación, se describe el conjunto de herramientas utilizadas para el desarrollo del prototipo.

2.3.1 Jupyter Notebook. Jupyter Notebook es una aplicación web de código abierto que permite crear y compartir documentos que contienen código en vivo, ecuaciones, visualizaciones y texto narrativo. Los usos incluyen: limpieza y transformación de datos, simulación numérica, modelado estadístico, visualización de datos, aprendizaje automático y mucho más.¹⁶

2.3.2 Librerías Python. Python¹⁷ es un lenguaje de programación interpretado, interactivo y orientado a objetos, que incorpora módulos, excepciones, tipificación dinámica, tipos de datos dinámicos de muy alto nivel y clases, además de combinar una potencia notable con una sintaxis muy clara. A continuación se muestra el conjunto de librerías utilizadas:

- **Numpy:** Numpy es la librería fundamental para la computación científica con Python. Se puede usar como un contenedor multidimensional eficiente de datos genéricos permitiendo que se integre sin problemas y rápidamente con una amplia variedad de bases de datos.¹⁸
- **Pandas:** Es una herramienta rápida, poderosa, flexible y fácil de usar para análisis y tratamiento de datos para el lenguaje de programación

¹⁶ JUPYTER. The jupyter notebook [En línea]. Project Jupyter, (16 de mayo de 2020). [Consultado: 18 de mayo de 2020]. Disponible en: <https://jupyter.org/>

¹⁷ PYTHON. Python [En línea]. [Consultado: 18 de mayo de 2020]. Disponible en: <https://docs.python.org/>

¹⁸ NUMPY. The fundamental package for scientific computing with Python [En línea]. [Consultado: 18 de mayo de 2020]. Disponible en: <http://www.numpy.org/>

Python. Su principal ventaja es que provee de amplias instalaciones para agrupar, fusionar y consultar datos.¹⁹

- **Scikit-learn:** Scikit-learn²⁰ contiene las herramientas necesarias para la minería y análisis de datos. Está construida sobre otras bibliotecas: Numpy, SciPy (Scientific Python) y Matplotlib. Incluye además herramientas como:

- **Clasificación:** Identifica a qué categoría pertenece un objeto.
- **Agrupación:** Asocia de forma automática objetos similares en conjuntos.
- **Selección de modelos:** Compara, valida y escoge parámetros y modelos.
- **Preprocesamiento:** Extrae características y normaliza.

- **NLTK:** NLTK²¹ es un conjunto de bibliotecas y programas para el procesamiento del lenguaje natural en Python. Incluye demostraciones gráficas y datos de muestra.

2.3.3 API Twitter. Para la recolección de datos se ha utilizado la API Streaming de Twitter²², la cual permite acceso a los tweets de perfiles de usuario y sus seguidores, quienes se han registrado usando autenticación y autorización OAuth. Estos datos se recogen en formato JSON.

¹⁹ PANDAS. Pandas [En línea]. [Consultado: 18 de mayo de 2020]. Disponible en: <https://pandas.pydata.org/>

²⁰ SCIKIT-LEARN. Machine learning in Python [En línea]. [Consultado: 18 de mayo de 2020] Disponible en: <https://scikit-learn.org/>

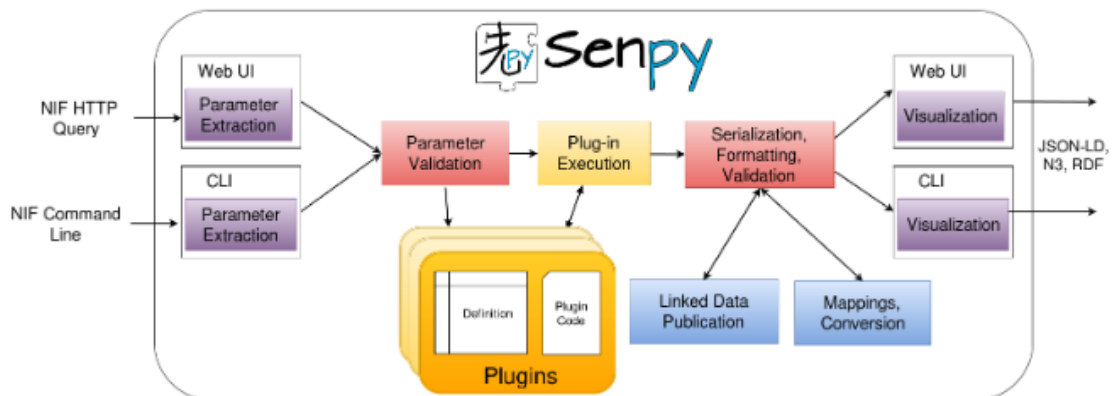
²¹ NLTK. NLTK 3.5 Documentation [En línea]. [Consultado: 18 de mayo de 2020]. Disponible en: <https://www.nltk.org/>

²² DEVELOPER. Publish and manage Tweets, and analyze Tweets data. [En línea]. [Consultado: 18 de mayo de 2020]. Disponible en: <https://developer.twitter.com/en/products/tweets>

2.3.4 Tweepy. Tweepy²³ es una librería de código abierto que permite comunicarlos con la API de Twitter a través de Python, posibilitando el acceso, conexión, desconexión, manejo de errores y filtrado con la API.

2.3.5 Senpy. Senpy²⁴ es un framework desarrollado por el grupo GSI de la Escuela Técnica Superior de Ingenieros de Telecomunicación en la Universidad Politécnica de Madrid, para el análisis de sentimientos y emociones, servicios que fueron implementados en un web service común para uso público.

Figura 1: Arquitectura Senpy API.



Fuente: Design and Development of a System for Sleep Disorder Characterization Using Social Media Mining. (Suárez, 2018).

La arquitectura de Senpy se basa en dos componentes:

- **Núcleo del Senpy:** Donde el servicio está construido y se encuentran todas las tareas comunes de un servicio web de análisis (Interacción de usuario, análisis de datos, logging, etc).

²³ TWEOPY. Tweepy documentation [En línea]. [Consultado: 18 de mayo de 2020]. Disponible en: <http://docs.tweepy.org>

²⁴ SENPY. Welcome to Senpy's documentation [En línea]. [Consultado: 18 de mayo de 2020]. Disponible en: <https://senpy.readthedocs.io/>

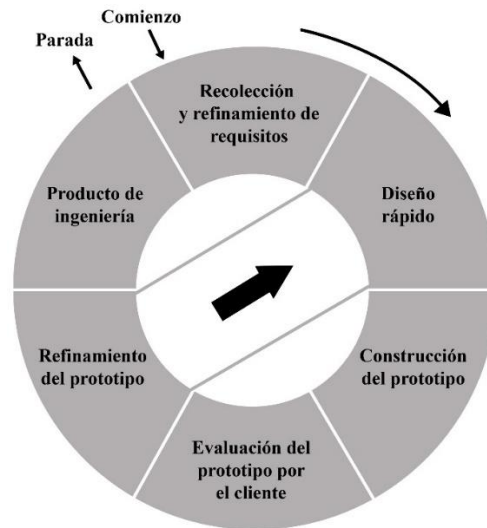
- **Plug-in Senpy:** Donde se encuentran los clasificadores de cada servicio por aparte.

En el caso particular de este proyecto, se hará uso del plug-in Sentiment MeaningCloud para determinar la polaridad de los trinos.

2.4 METODOLOGÍA

El presente proyecto está elaborado bajo una metodología de desarrollo de sistemas, que permite la división del trabajo en diferentes fases que contienen actividades, con la intención de mejora en la planificación y gestión llamada modelo de desarrollo evolutivo (ver Figura 2).

Figura 2: Fases del desarrollo evolutivo.



Fuente: Zachman, John A. El modelado de las empresas: La arquitectura de Zachman. Zachman Institute for Framework Advancement. Estados Unidos, 1999.

Este desarrollo se basa en la creación de versiones parciales del programa que se está desarrollando. Un prototipo comúnmente expone aspectos o características del software final pero que pueden cambiar durante el desarrollo e incluso no ser iguales en el producto final. Esto facilita que se evalúe rápidamente el prototipo para detectar fallas y agregar cambios.²⁵

De esta manera, se puede tener idea de las estimaciones iniciales y realizar comparaciones, para comprobar que el prototipo cumpla con las especificaciones planteadas y así evaluar el cumplimiento de los plazos pactados inicialmente. Esto es útil cuando se conocen los objetivos generales pero se desconoce la adaptabilidad de un sistema operativo o la eficacia de algún algoritmo.

²⁵ COLABORADORES DE WIKIPEDIA. Modelo de prototipos [En línea]. Wikipedia, la enciclopedia libre, 2018. [Consultado: 18 de mayo de 2020]. Disponible en: https://es.wikipedia.org/wiki/Modelo_de_prototipos

3. METODOLOGÍA PARA LA CLASIFICACIÓN DE TEXTOS

Para la clasificación de las publicaciones Twitter, se diseñó y llevó a cabo una metodología compuesta por dos fases principales, como lo son la construcción del modelo y las predicciones realizadas por el mismo, las cuales se describen brevemente en este capítulo. Los capítulos 4 y 5 presentarán en detalle cada una de estas fases.

FASE 1: CONSTRUCCIÓN DEL MODELO

Como se puede observar en la Figura 3, esta primera fase consta de 4 etapas en las que se recolectan y preparan los datos, se entrenan los algoritmos de clasificación, se mide su rendimiento y finalmente se visualizan los resultados obtenidos.

Figura 3: Fase 1 - Construcción del modelo

Fase 1: Construcción del modelo



Recolección y pre procesamiento de tweets: Inicialmente se definen los parámetros de recolección de los datos de entrenamiento para Colombia y se solicita acceso a la API de Twitter. Se procede a realizar el descargue y guardado de los de los tweets y posteriormente se preparan los datos mediante los procesos de limpieza y tokenización y estemizado.

Extracción de características: Una vez los tweets han sido pre procesados, se realiza un filtrado por palabras clave relacionadas con síntomas de trastorno depresivo para identificar los trinos asociados y el respectivo síntoma. Se procede a asignar un valor numérico para indicar la polaridad de cada oración mediante el análisis de sentimientos, se realiza una clasificación manual al 70% de los datos que serán usados como entrenamiento para los clasificadores y el 30% se destinará para el testeo de los mismos y finalmente la vectorización de los textos.

Selección y entrenamiento de clasificadores: En esta parte del proceso se seleccionan los algoritmos de clasificación a utilizar y se definen los conjuntos de referencia y prueba para llevar a cabo el entrenamiento de los clasificadores.

Evaluación de clasificadores y resultados: Finalmente se definen las métricas con las cuales se evaluarán los clasificadores y se presentan los resultados obtenidos de dicha medición, la cantidad de posibles casos identificados por cada algoritmo y los síntomas y palabras más recurrentes durante esta fase de construcción del modelo.

FASE 2: PREDICCIONES DEL MODELO

Para esta fase del procedimiento se realizan a través de una tubería de datos, y de manera similar, las etapas descritas en la construcción del modelo. Sin embargo difiere en procesos como la extracción de características donde ya no se ejecuta la clasificación manual y se elimina una de sus etapas pues ya no es necesaria la selección, entrenamiento y evaluación de clasificadores (ver Figura 4).

Figura 4: Fase 2 - Predicciones del modelo

Fase 2: Predicciones del modelo entrenado



Recolección y pre procesamiento: Para iniciar la fase de predicciones del modelo entrenado se realiza el descargue y guardado de los tweets para la ciudad de Medellín, y se preparan los datos mediante los procesos de limpieza y tokenización y estemizado.

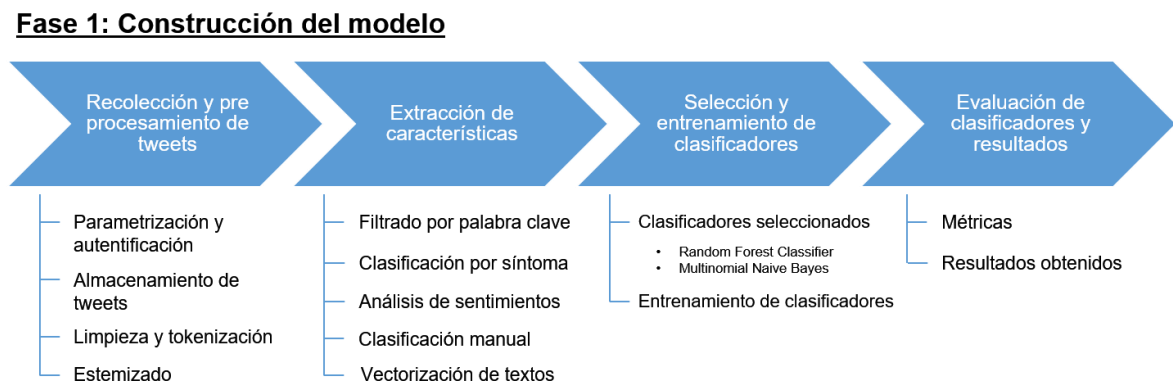
Extracción de características: Se procede a identificar los síntomas depresivos asociados a cada uno de los tweets, se lleva a cabo el análisis de sentimientos para determinar su polaridad y finalmente se vectorizan los textos.

Uso del modelo y visualización de resultados: Para concluir con el procedimiento se hace uso del modelo entrenado para encontrar posibles casos de depresión y se presentan los resultados obtenidos.

4. CONSTRUCCIÓN DEL MODELO

En este capítulo se describe en detalle la fase 1: Construcción del modelo, presentando las etapas y actividades diseñadas y realizadas, así como el código fuente desarrollado. La Figura 5 presenta las actividades para cada una de las etapas de la fase 1.

Figura 5: Etapas de la fase 1 construcción del modelo



4.1 RECOLECCIÓN Y PRE PROCESAMIENTO DE TWEETS

La captura de las publicaciones de la red social Twitter llamadas tweets (trinos en español), fue realizada desde el 25 de octubre hasta el 18 de noviembre de 2019 a través de la API Streaming de Twitter y la librería Tweepy que nos permite recolectar trinos en tiempo real. De esta forma se recogieron 100.000 tweets de usuarios ubicados en Colombia.

Los tweets recolectados para este proyecto debían contener tres características importantes:

- Los tweets deben estar en idioma español.

- No pueden ser re-tweets (es decir un tweet re postado de un usuario diferente).
- Todos los tweets deben pertenecer a usuarios que se encuentren dentro de la zona de interés a analizar.

4.1.1 Parametrización y autenticación. Para llevar a cabo este proceso se realiza el importe de librerías y se establecen los parámetros de recolección como el idioma de los tweets, las coordenadas y radio de búsqueda, el rango de días de recolección y las columnas del conjunto de datos (*dataset*) en el cual se guardará la información. A continuación se realiza una autenticación por medio de una *API key* y un *token* de acceso provisto por la API de Twitter, con el fin de obtener autorización para el descargue de los tweets. Ver Figura 6.

Figura 6: Sección de código - Parametrización y autenticación

DETECCIÓN TEMPRANA DE POSIBLES CASOS DE DEPRESIÓN EN COLOMBIA MEDIANTE EL ANÁLISIS DE TENDENCIAS EN TWITTER

```
import numpy as np
import pandas as pd
import tweepy as tp
import nltk
import matplotlib.pyplot as plt
```

Parametrización

```
NUMERO_TWITS = 5000
LENGUAJE = "es"
LOCALE = "es"
NOMBRES_USUARIO = False
GEOCODE = "4.536307,-75.6723751,1.6km" # Latitud, Longitud, radio [km/mi]
FECHA_DESDE = '' #yyyy-mm-dd
FECHA_HASTA = '' #yyyy-mm-dd
COLUMNAS = ['fecha', 'hora', 'texto', 'coordenadas', 'nombre_usuario', 'municipio', 'codigo_pais', 'pais']
APIKEY_MC = "e5832ac9c6b17bfb8b478d250b64da45"
```

Autenticación

```
import tweepy
auth = tweepy.OAuthHandler("qdfjJrIJVoioqOQsbM1sJGSZE", "uBMKxLvIdFQGLX7cBx6Gwcgd1Kp4p57wQlJbGr3NBuyp0cvSd7")
auth.set_access_token("1138469115386114048-z33QhHaFktUGRneFJ3bMEiV9XwpCxH", "LEixc2pAImpEpDlGS9Vud19yJi0RyIMetsXZaRTOTyNvG")
api = tweepy.API(auth, wait_on_rate_limit=True)
```

4.1.2 Almacenamiento de tweets. Habiendo autorizado el acceso se definen las funciones *recolectar* y *generar array*, en las cuales se realiza el descargue de trinos tomando los parámetros mencionados en la sección 3.1.1. Estos se guardarán en arreglos de datos para formar el *dataset* en formato *.csv* con el nombre *dataset_tweets_pf_1*, que contendrá información como hora y fecha de la publicación y el texto del tweet, y datos visibles autorizados por el usuario como coordenadas correspondientes a su ubicación, nombre de usuario, ciudad, código del país y país (ver Figura 7).

Figura 7: Sección de código - Almacenamiento de tweets

Recolección de datos

```
# Método recolector de datos
def recolectar():
    twits = []
    cadena_búsqueda = '-filter:retweets -filter:replies '
    if FECHA_DESDE != "":
        cadena_búsqueda += 'since:' + FECHA_DESDE + ' '
    if FECHA_HASTA != "":
        cadena_búsqueda += 'until:' + FECHA_HASTA
    print (cadena_búsqueda)
    for tweet in tweepy.Cursor(api.search,
                               q = cadena_búsqueda, #No retweets
                               lang = LENGUAJE,
                               locale = LOCALE,
                               geocode = GEOCODE,
                               show_user = NOMBRES_USUARIO).items(NUMERO_TWITS):
        twits.append(tweet)
    return twits
```

```
def generar_array(lista_twits):
    import numpy as np
    array = np.ndarray((len(lista_twits),8),dtype=object)
    for i in range(len(lista_twits)):
        fecha = lista_twits[i].created_at.date()
        hora = lista_twits[i].created_at.time()
        array[i,0] = fecha
        array[i,1] = hora
        array[i,2] = lista_twits[i].text
        array[i,3] = lista_twits[i].coordinates
        array[i,4] = lista_twits[i].user.screen_name
        #print (lista_twits[i].place)
        if lista_twits[i].place != None:
            array[i,5] = lista_twits[i].place.name
            array[i,6] = lista_twits[i].place.country_code
            array[i,7] = lista_twits[i].place.country
        else:
            array[i,5] = None
            array[i,6] = None
            array[i,7] = None
    return array
```

```
var = recolectar()
array = generar_array(var)
```

```
dataset = pd.DataFrame(data=array, columns=COLUMNAS)
dataset.to_csv('..\datos\dataset_tweets_pf_1.csv', index = False)
```

4.1.3 Tokenización y limpieza de tweets. En esta fase se toman todos los tweets para limpiarlos antes de ser procesados. Para esto se desarrolla un módulo de tratamiento de datos donde en primera instancia se eliminan signos de puntuación, caracteres extraños, emoticones, urls y palabras que no aporten información significativa acerca del contenido que se quiere analizar. Para llevar a cabo esta tarea se define un conjunto de palabras irrelevantes contenidas en el listado provisto por la librería NLTK como pronombres, artículos, preposiciones y demás, a través de *stopwords.words* y *string.punctuation*. En este conjunto estarán palabras como: *en, sus, fue, tú, hasta, también, han, tuvieron*, entre otras y signos de puntuación y caracteres especiales como el arroba, el signo pesos o el numeral.

Conjuntamente se lleva a cabo el proceso de *tokenización* gracias a la función *word_tokenize* (ver Figura 8) en el cual, dada una secuencia de caracteres y una unidad de documento definida, esta se divide en partes llamadas *tokens*. Para este caso en específico los *tokens* serán palabras individuales propias de cada tweet.

4.1.4 Estemizado. Adicionalmente se realiza el proceso de estemizado, el cual es un proceso lingüístico que consiste en encontrar la correspondiente raíz o forma base de una palabra, entendiéndose por forma base como la raíz invariable de palabras relacionadas por su forma, y para lo cual se eliminan sus terminaciones. El resultado de este proceso es la forma que por convención se acepta como el representante de todas las formas flexionadas de la misma palabra, la cual no es necesariamente otra palabra.

Para llevar a cabo esta tarea se define la función *transformar*, donde gracias al algoritmo “*SnowballStemmer*” de la librería NLTK que contiene las raíces de palabras en español, se aplica el proceso de estemizado. De esta manera palabras como: *desesperado, lágrimas, soledad y deprimido*, se convertirán en: *desesper, lagrim, soled y deprim*. En ocasiones varias palabras comparten su misma forma

base, como es el caso de: resignación, resignado, resignadas, cuya raíz invariable es *resign*.

Tanto el proceso de limpieza y *tokenización* como el de *estemizado* se agrupan finalmente en la función *procesar* dentro de la clase denominada "TLN".

Figura 8: Sección de código - Módulo de tratamiento de textos

```
from nltk import word_tokenize
from nltk.corpus import stopwords
from nltk.stem.snowball import SnowballStemmer
import string

class TLN:
    descripcion = "Clase prototipo para tratamiento básico de textos"
    version = "0.1"

    def __init__(self, idioma="english"):

        self.__IDIOMA = ""
        self.__P_IRRELEVANTES = []
        self.cambiarIdioma(idioma)

    def cambiarIdioma(self, idioma):
        self.__IDIOMA = idioma
        self.__P_IRRELEVANTES = stopwords.words(self.__IDIOMA) + list(string.punctuation)

    def tokenizar(self, texto):
        return [palabra for palabra in word_tokenize(texto.lower()) if not palabra in self.__P_IRRELEVANTES]

    def transformar(self, lista_palabras):
        palabras_transformadas = []
        stemmer = SnowballStemmer(self.__IDIOMA)
        for palabra in lista_palabras:
            palabras_transformadas.append(stemmer.stem(palabra))
        return palabras_transformadas

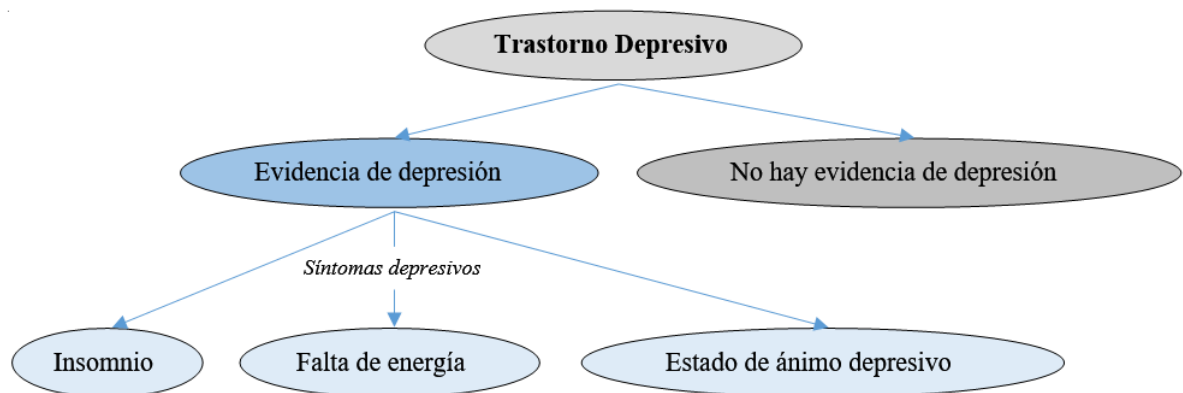
    def procesar(self, texto):
        fase_1 = self.tokenizar(texto)
        return self.transformar(fase_1)
```

4.2 EXTRACCIÓN DE CARACTERÍSTICAS

4.2.1 Filtrado por palabra clave. Diversos estudios han mostrado que los usuarios expresan y discuten acerca de su salud mental a través de redes sociales, y que estos datos pueden predecir un potencial riesgo de depresión. Los factores de riesgo y experiencias varían, sin embargo se encuentran síntomas que persisten y tienden a ser iguales en la mayoría de la población, como lo son el estado de ánimo deprimido, fatiga o pérdida de energía y dificultad para dormir.²⁶

En el procesamiento de los tweets esta clasificación se realiza gracias al esquema “*Síntomas Depresivos y Factores Psicosociales Asociados con la Depresión*” (ver Figura 9). Esta clasificación permite no sólo determinar si el mensaje tiene una connotación depresiva o no, sino que además proporciona información acerca de uno o más síntomas asociados.

Figura 9: Esquema de trastorno depresivo.



Fuente: Towards Automatically Classifying Depressive Symptoms from Twitter Data for Population Health (Mowery et al., 2016).

²⁶ (Mowery et al, 2016)

Bajo este esquema se determina si un tweet muestra síntomas de un posible caso de depresión y se adoptan palabras claves asociadas a la depresión con base en el Manual Diagnóstico y Estadístico de los Trastornos Mentales (Asociación Americana de Psiquiatría, 2013). En la Tabla 1 se indican las palabras relacionadas a cada uno de los síntomas.

Tabla 1: Palabras clave relacionados con tweets depresivos.

CATEGORÍA	PALABRAS RELACIONADAS
Insomnio	Insomnio, dormir.
Falta de energía	Fatiga, cansancio.
Estado de ánimo depresivo	Abandonado, llorar, depresión, devastado, desilusión, soledad, arrepentimiento, remordimiento, resignación, tristeza, impotencia, infeliz, melancólico, inútil, derrotado, desesperación, decepcionar, insatisfacción, aflicción, lastimado, nostálgico, perdedor, sufrimiento, lagrimas, triste.

Fuente: Towards Automatically Classifying Depressive Symptoms from Twitter Data for Population Health (Mowery et al., 2016).

Para llevar a cabo el filtrado de tweets por palabra clave, es necesario aplicar la clase “TLN” mencionada en la sección 4.1 a cada conjunto de palabras relacionadas con cada síntoma, los cuales se llamarán *PC_INSOMNIO*, *PC_ENERGÍA* y *PC_DEPRESION* (Ver Figura 10). De esta manera, se podrá realizar una comparación que permita identificar los trinos que contienen una o más palabras clave y descartar los que no.

Al realizar el proceso de filtrado, los tweets resultantes contendrán al menos una palabra asociada a un síntoma depresivo y sus textos estarán tokenizados y estemizados, lo cual facilitará que se lleven a cabo los demás procesos como vectorización y posteriormente el entrenamiento de algoritmos de clasificación, este dataset recibe el nombre de: *dataset_procesado*. Sin embargo procesos como el análisis de sentimientos o la clasificación manual se verán entorpecidos dada la dificultad para entender el texto. Es por esto que paralelamente se efectúa la creación de otro dataset idéntico con texto sin pre procesar el cual se llamará: *dataset_filtrado*.

Figura 10: Sección de código - Filtrado de tweets

```

from Clases.TLN import TLN

tln = TLN("spanish")
datos = dataset_global.values
datos_copia = np.copy(datos)
PC_INSOMNIO = ['insomnio', 'dormir']
PC_ENERGIA = ['fatiga', 'cansancio']
PC_DEPRESION = ['abandonado', 'llorar', 'depresión', 'desilusión', 'devastado', 'soledad', 'arrepentimiento',
               'remordimiento', 'resignación', 'tristeza', 'impotencia', 'infeliz', 'melancólico', 'inútil', 'derrotado',
               'desesperación', 'desilusión', 'decepcionar', 'insatisfacción', 'aflicción', 'lastimado', 'nostálgico',
               'perdedor', 'sufrimiento', 'lagrimas', 'triste']
PALABRAS_CLAVE = PC_INSOMNIO + PC_ENERGIA + PC_DEPRESION
PC_INSOMNIO_BASE = tln.transformar(PC_INSOMNIO)
PC_ENERGIA_BASE = tln.transformar(PC_ENERGIA)
PC_DEPRESION_BASE = tln.transformar(PC_DEPRESION)
PALABRAS_CLAVE_BASE = PC_INSOMNIO_BASE + PC_ENERGIA_BASE + PC_DEPRESION_BASE

twits_procesados = []
twits_filtrados = []
for i in range(len(datos)):
    twit_procesado = tln.procesar(datos[i,2])
    for palabra in twit_procesado:
        if palabra in PALABRAS_CLAVE_BASE:
            datos_copia[i,2] = ' '.join([str(elem) for elem in twit_procesado])
            datos[i,2] = tln.limpiar(datos[i,2])
            twits_procesados.append(datos_copia[i])
            twits_filtrados.append(datos[i])
            break

datos_procesados = np.asarray(twits_procesados, dtype=object)
datos_filtrados = np.asarray(twits_filtrados, dtype=object)

dataset_filtrado = pd.DataFrame(data=datos_filtrados, columns=COLUMNAS)
dataset_procesado = pd.DataFrame(data=datos_procesados, columns=COLUMNAS)

```

4.2.2 Clasificación por síntoma. Para concluir la clasificación por síntoma de los tweets, se propone un sistema de numeración octal de asignación de valores, para identificar el tipo de síntoma depresivo que se está presentando en el tweet. Es decir, que cada palabra clave que se encuentre en el texto significará un uno o de lo contrario un cero y formará un código que estará asociado a un síntoma o la combinación entre ellos.

Tabla 2: Código de clasificación de síntomas.

Código	Número	Etiqueta
000	0	Error
001	1	Depresivo
010	2	Falta de energía
011	3	Falta de energía y depresivo
100	4	Insomnio
101	5	Insomnio y depresivo
110	6	Insomnio y falta de energía
111	7	Todos los anteriores

Para etiquetar cada uno de los tweets se define la función *etiquetadoOctal*, dentro de la cual se usará una variable interruptor denominada *bandera*, que cambiará su valor al encontrarse con una palabra clave en cada uno de las temáticas definidas (ver Figura 11).

Figura 11: Sección de código – Clasificación por temática

```
dataset_procesado = pd.read_csv('../datos/dataset_pf_procesado_FASE2_v1.csv')
dataset_filtrado = pd.read_csv('../datos/dataset_pf_filtrado_v3.csv')
```

```
datos_procesados = dataset_procesado.values
```

```
def etiquetadoOctal(twit):
    resultado = ''
    def chequear(bandera):
        return '1' if bandera == True else '0'
    bandera = False

    for palabra in twit:
        if palabra in PC_INSOMNIO_BASE:
            bandera = True
            break
    resultado += chequear(bandera)
    bandera = False

    for palabra in twit:
        if palabra in PC_ENERGIA_BASE:
            bandera = True
            break
    resultado += chequear(bandera)
    bandera = False

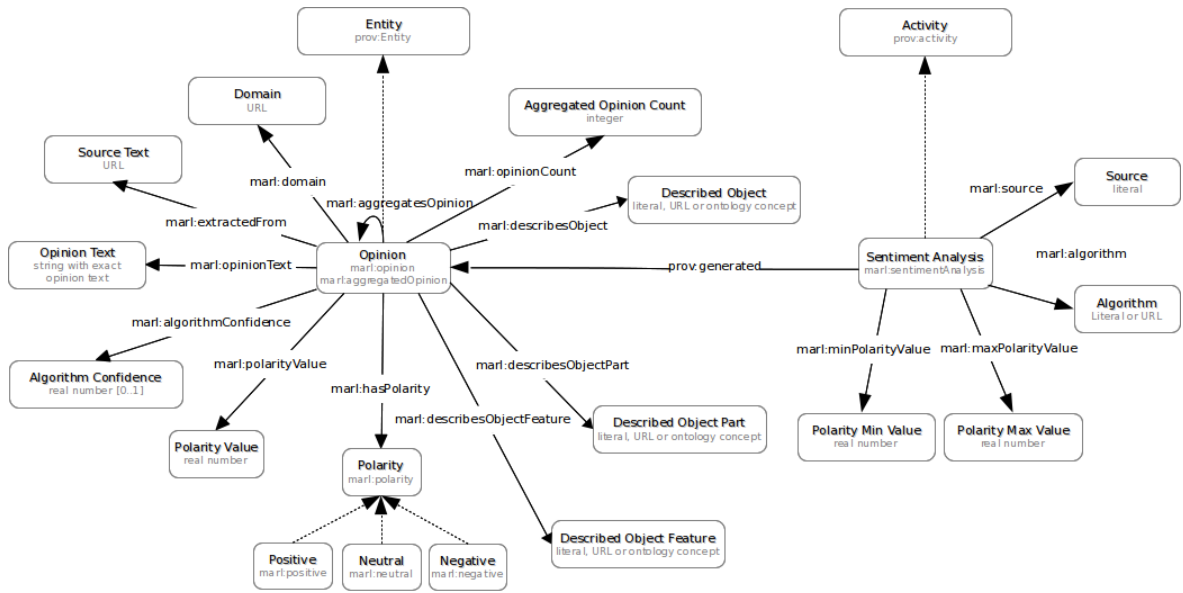
    for palabra in twit:
        if palabra in PC_DEPRESION_BASE:
            bandera = True
            break
    resultado += chequear(bandera)
    bandera = False

    return int(resultado,2)
```

4.2.3 Análisis de sentimientos. En esta parte del proceso se analizarán los textos para encontrar rasgos asociados a emociones positivas o negativas, asignando un valor que representa la polaridad del texto, es decir, el carácter positivo, negativo o neutral de un tweet. Para esto, los trinos son analizados a través de Senpy API haciendo uso del plug-in Sentiment MeaningCloud, el cual utiliza técnicas de procesamiento del lenguaje natural para detectar la polaridad asociada a palabras y oraciones completas en el texto, teniendo en cuenta su estructura gramatical bajo la ontología Marl (Ver Figura 12).

Este esquema de datos estandarizados desarrollado por la Universidad Politécnica de Madrid, se usa para describir opiniones subjetivas expresadas en la web o en sistemas de información en particular, etiquetando así un texto como positivo o negativo

Figura 12: Ontología Marl.



Fuente: Especificación de la Ontología Marl (Consultado el 21 de febrero del 2020)
 Disponible en: <http://www.gsi.dit.upm.es/ontologies/marl/>

Bajo este modelo ontológico, los tweets son enviados a través de un servidor web de demostración donde son analizados por Senpy API, quien finalmente emite una valoración que describe la emoción presentada (ver Figura 13).

Figura 13: Captura de pantalla ejemplo de asignación de polaridad

Enter the text you want to analyze or select one of the pre-defined examples:

Regular Tweet

Hoy me siento muy triste

Select the plugin.

sentiment-meaningcloud +

Sentiment analysis with meaningCloud service. To use this plugin, you need to obtain an API key from meaningCloud signing up here: <https://www.meaningcloud.com/developer/login>

```
{
  "@context": "http://senry.gsi.uqm.es/api/contexts/YX8qLz9hb8dvPX0lkn8obWudC1t7uFvsh5nY2xvdwQmeT11b3k7MjBt7SUyWHJpZ3V5S8byUyM611",
  "@type": "Results",
  "entries": [
    {
      "@id": "prefix:",
      "@type": "Entry",
      "language_detected": "es",
      "marl:hasOpinion": [
        {
          "@id": "Opinion0",
          "@type": "Sentiment",
          "marl:hasPolarity": "marl:Negative",
          "marl:opinionCount": 1,
          "marl:polarityValue": -1,
          "prov:wasGeneratedBy": "endpoint:plugins/sentiment-meaningcloud_1.1"
        },
        {
          "@id": "Opinion1",
          "@type": "Sentiment",
          "marl:aggregatesOpinion": null,
          "marl:hasPolarity": "marl:Negative",
          "marl:polarityValue": -1,
          "nif:anchorOf": "Hoy me siento muy triste",
          "nif:beginIndex": 8,
          "nif:endIndex": 23,
          "prov:wasGeneratedBy": "endpoint:plugins/sentiment-meaningcloud_1.1"
        }
      ],
      "nif:isString": "Hoy me siento muy triste",
      "onyx:hasEmotionSet": []
    }
  ]
}
```

Fuente: Senpy Playground. (Consultado el 21 de febrero del 2020) Disponible en: <http://senpy.gsi.upm.es/>

Para hacer uso del servidor de demostración se crea una variable *endpoint* y una función auxiliar *query* para simplificar las consultas e imprimir los resultados. Se define la función *extraer* en donde Senpy devolverá un objeto de tipo *senpy:Results* con una lista de entradas en formato json. Cada entrada contiene el texto original y demás propiedades proporcionadas por el complemento, en este caso un valor indicando la polaridad del tweet a través de *marl:polarityValue*. El dataset utilizado para este propósito será el *dataset_filtrado* (ver Figura 14).

Figura 14: Sección de código - Análisis de sentimientos

Procesamiento por Senpy

Parámetros y métodos necesarios

```
ENDPOINT = 'http://senpy.gsi.upm.es/api'
import requests
import json
from IPython.display import Code
from tqdm import tqdm

def query(endpoint, **kwargs):
    '''Query a given Senpy endpoint with specific parameters, and prettify the output'''
    res = requests.get(endpoint,
                       params=kwargs)
    if res.status_code != 200:
        raise Exception(res)
    return res.text

def extraer(base):
    #print (json)
    #return json['entries'][0]['marl:hasOpinion'][0]['marl:hasPolarity']
    return json.loads(base)['entries'][0]['marl:hasOpinion'][0]['marl:polarityValue']

i = 0
for twit in tqdm(datos_filtrados[i:,2]):
    datos_procesados[i,8] = extraer(query(f'{ENDPOINT}/sentiment-meaningcloud', input=twit, language='es', apiKey=APIKEY_MC))
    i += 1
```

4.2.4 Clasificación manual de los tweets. Para que los algoritmos de clasificación cumplan con la tarea de predecir el riesgo de un posible caso de depresión, estos deben ser entrenados con base en información previamente evaluada por el ser humano. Esta evaluación estará consignada en la columna *t1* del *dataset* y se realiza bajo los siguientes criterios:

- Usuario que presenta síntomas de un posible caso de depresión: La publicación realizada por el usuario tiene connotación negativa y expresa claros síntomas de depresión, o podría resultar en un caso futuro. Se considera necesario que un profesional de la salud mental analice más a fondo el caso.
- Usuario que no presenta síntomas de un posible caso de depresión: La publicación realizada por el usuario no presenta ningún síntoma de depresión.

4.2.5 Vectorización de textos. Antes de realizar el entrenamiento de los algoritmos de clasificación, se debe preparar el texto realizando una extracción de características, o en otras palabras, otorgarle un formato para que estos puedan funcionar adecuadamente. Más específicamente, los textos deben convertirse en vectores de longitud fija en los cuales aparece la frecuencia de repetición de las palabras que allí se encuentran (ver Figura 15). A este proceso se le denomina vectorización y el modelo escogido para realizarse será el de “bolsa de palabras”, el cual es una representación del texto que describe la aparición de palabras en un documento pero no especifica su orden.²⁷

²⁷ ZHOU, Victor. A simple explanation of the bag of words model [En línea]. Towards data science, 2019. [Consultado: 8 de mayo de 2020]. Disponible en: <https://towardsdatascience.com/a-simple-explanation-of-the-bag-of-words-model-b88fc4f4971>

Figura 15: Ejemplo de creación de bolsa de palabras

Tweet:	Hoy	me	siento	muy	triste
Hoy	1	0	0	0	0
me	0	1	0	0	0
siento	0	0	1	0	0
muy	0	0	0	1	0
triste	0	0	0	0	1

La creación de este modelo de bolsa de palabras en el presente proyecto, dio lugar a una matriz de densidad que contiene los tweets vectorizados y las columnas de clasificación manual y análisis de sentimientos por parte de Senpy API.

Para ello, se hará uso de la librería *scikit-learn* con las funciones *CountVectorizer* y *TfidfTransformer* en el proceso de vectorización y para calcular el número de elementos que allí se encuentran, procesos que se llevarán a cabo en el *dataset_procesado* (ver Figura 16).

Figura 16: Sección de código - Vectorización de los textos

Creación y configuración del Bag of Words (BoW)

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer as tf
dataset = pd.read_csv('../datos/dataset_pf_procesado_FASE2_v1.csv')
vectorizer = CountVectorizer()
datos_main = dataset.values
cuenta = vectorizer.fit_transform(datos_main[:,2])
```

Creación de la matriz de densidad

La matriz de densidad incluirá el BoW y las columnas de datos adicionales (que contiene el target clasificado manualmente y la columna de polaridad otorgada por el Senpy)

Al final, la matriz debe de quedar del tamaño $N_{columnas_BoW} + 1$

```
smatrix = vectorizer.transform(datos_main[:,2])
smatrix.todense()
t = tf()
t.fit(smatrix)
tf_matrix = t.transform(smatrix)
mat = tf_matrix.todense()
mat.shape
```

```
# Adición de la columnas adicionales
adicionales = datos_main[:,8:9]
mat = np.hstack((mat,adicionales))
```

4.3 SELECCIÓN Y ENTRENAMIENTO DE CLASIFICADORES

La última parte del proceso de creación del modelo corresponde a la selección y entrenamiento de los clasificadores a usar. Estos algoritmos deberán ser capaces de clasificar los trinos provenientes de la red social Twitter e identificar posibles casos de depresión.

4.3.1 Clasificadores seleccionados. Dado el problema de clasificación de textos al cual se enfrentarán los algoritmos de aprendizaje automático, autores como Nadeem²⁸ y Mowery²⁹ sugieren el uso de algoritmos supervisados para realizar esta tarea. Ambos estudios exponen el rendimiento de diferentes algoritmos que trabajan a partir de datos etiquetados entre los cuales se destacan *Random Forest Classifier* y *Multinomial Naive Bayes*, los cuales harán uso del etiquetado manual descrito en la sección 4.3 para posteriormente clasificar nuevos textos. Para tal fin se empleará la librería Scikit-learn que cuenta con dichos algoritmos.

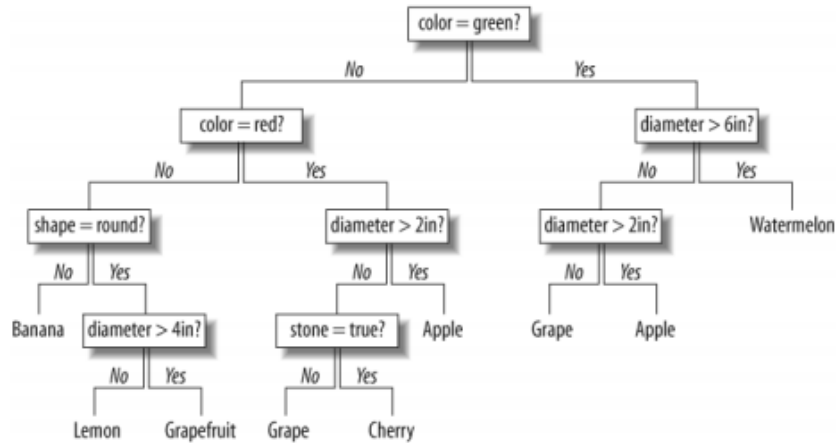
4.3.1.1 Random Forest Classifier (RaF): Random Forest Classifier (Bosques aleatorios en español) es un meta estimador que se ajusta a varios clasificadores de árboles de decisión en varias sub muestras del conjunto de datos y utiliza el promedio para mejorar la precisión predictiva y controlar el sobreajuste. Su estructura es la de un grafo dirigido en forma de árbol compuesto por un conjunto de reglas extraídas a partir de las características de los datos de entrenamiento y que se aplican de manera sucesiva a la hora de predecir a qué clase pertenece un nuevo ejemplo (ver Figura 17). En general, un árbol de decisión está formado por nodos y líneas que unen dichos nodos, comenzando en la raíz y terminando en varios con las posibles clasificaciones que se pueden establecer a una muestra dada.³⁰

²⁸ NADEEM, Op. Cit., p.5

²⁹ MOWERY, Op. Cit., p.185

³⁰ SOBRINO, Op. Cit., p. 44

Figura 17: Ejemplo de clasificación a través de un árbol de decisión



Fuente: SOBRINO, José. Árboles de decisión [Imagen]. Análisis de sentimientos en Twitter, 2018. p. 44. [Consultado: 8 de mayo de 2020]. Disponible en: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/81435/6/jsobrinosTFM0618memoria.pdf>

4.3.1.2 Multinomial Naive Bayes (MNB): El clasificador bayesiano ingenuo multinomial es un modelo probabilístico de aprendizaje automático que asume que cada componente (palabra) del texto es totalmente independiente de los otros componentes (palabras). Así, en vez de ver al texto como una serie de palabras conectadas de manera lógica, el texto se analiza como un conjunto de palabras independientes entre sí. Este punto de vista se considera “ingenuo” dado que es evidente que en un texto las palabras tienen relación entre ellas. A pesar de esta característica el rendimiento del método es destacable y es de fácil aplicación, dado su simple planteamiento, y de ahí su amplio uso en muchos problemas de la vida real. Para la tarea de clasificación en este caso asume un enfoque multinomial basado en el teorema de Bayes.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Usando el teorema de Bayes, podemos encontrar la probabilidad de que ocurra A, dado que B ha ocurrido. Aquí, B es la evidencia y A es la hipótesis. La suposición hecha aquí es que las características son independientes. Es decir, la presencia de una característica particular no afecta a la otra. Por eso se le denomina ingenuo.³¹

4.3.2 Entrenamiento de clasificadores. Para llevar a cabo el entrenamiento de los clasificadores, como se observa en la Figura 18, se definen los conjuntos de prueba y los datos de referencia para implementar y entrenar los modelos predictivos. Así, en el conjunto *x_train* estarán todas las variables independientes usadas para entrenar los modelos, en *y_train* las variables dependientes predichas, *x_test* corresponderá a las variables independientes que no fueron usadas para entrenamiento y serán usadas para predecir, y finalmente *y_test* en el que se encuentran los resultados predichos.

Figura 18: Sección de código - Entrenamiento de los clasificadores

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import MultinomialNB
```

Implementación de modelos predictivos y entrenamiento de clasificadores

```
# Creación de los conjuntos de prueba y entrenamiento
lon_datos = len(datos_main)
porc_train = int(lon_datos*0.7)
x_train = mat[:porc_train]
y_train = datos_main[:porc_train,9].astype('int')
x_test = mat[porc_train:]
y_test = datos_main[porc_train:,9].astype('int')
```

```
CLASIFICADORES = {
    'RaF' : {
        'Clasificador' : RandomForestClassifier(random_state=np.random.seed(8888)),
        'Cm' : 0,
        'y_predicts' : 0
    },
    'MNB' : {
        'Clasificador' : MultinomialNB(alpha=0, fit_prior=False),
        'Cm' : 0,
        'y_predicts' : 0
    }
}
for nombre_c in CLASIFICADORES:
    CLASIFICADORES[nombre_c]['Clasificador'].fit(x_train,y_train)
```

³¹ GHANDI, Rohith. Naïve Bayes Classifier [En línea]. Towards data science, 2018. [Consultado: 8 de mayo de 2020]. Disponible en: <https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c>

4.4 EVALUACIÓN DE CLASIFICADORES Y RESULTADOS

4.4.1 Métricas. Luego de haber entrenado y usado los clasificadores, se realizan mediciones con el fin de determinar la calidad de las predicciones hechas por cada algoritmo. Para ello se tomará como connotación positiva el hecho de identificar un posible caso de depresión y con base en esto se definen una serie de conceptos y métricas a utilizar:

- **Verdaderos positivos (VP):** predichos correctamente con una connotación positiva (Depresivos predichos como depresivos).
- **Verdaderos negativos (VN):** predichos correctamente con una connotación negativa (No depresivos predichos como no depresivos).
- **Falsos positivos (FP):** predichos de manera incorrecta con connotación positiva (No depresivos que fueron predichos como depresivos).
- **Falsos negativos (FN):** predichos de manera incorrecta con connotación negativa (Depresivos que fueron predichos como no depresivos).

Estas definiciones permiten calcular las siguientes métricas:

- **Accuracy:** Mide el porcentaje de predicciones correctas en relación con el número total de predicciones.

$$\frac{VP + VN}{VP + VN + FP + FN}$$

- **Precision (P):** Mide el porcentaje de predicciones positivas correctas en relación al total de predicciones positivas.

$$\frac{VP}{VP + FP}$$

- **Recall (R):** Mide el porcentaje de predicciones positivas correctas en relación con el total de valores positivos esperados.

$$\frac{VP}{VP + FN}$$

- **F1 Score:** Es la medida de precisión que tiene un test, teniendo en cuenta las métricas de Recall (R) y Precision (P).

$$2 \times \frac{R \times P}{R + P}$$

Estas métricas se importan a través de la librería scikit-learn. Algunas de ellas como *precision* y *recall* requieren valores de decisiones binarias. Se crea la matriz de confusión, en donde cada fila representará las instancias en una clase predicha y cada columna las instancias de una clase real, dando a conocer el total de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos obtenidos. Para llevar a cabo la creación de la matriz y de las métricas se tendrán en cuenta los conjuntos *y_test* y *y_predicts* (ver Figura 19).

Figura 19: Sección de código – Métricas del modelo entrenado

```
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
from sklearn.metrics import confusion_matrix

for nombre_c in CLASIFICADORES:
    print ("===", nombre_c, "===")

    #Predicciones
    y_predicts = CLASIFICADORES[nombre_c]['Clasificador'].predict(x_test)
    CLASIFICADORES[nombre_c]['y_predicts'] = y_predicts
    #Conteo
    cm = confusion_matrix(y_test, y_predicts)
    CLASIFICADORES[nombre_c]['Cm'] = cm
    v_positivos, f_positivos, f_negativos, v_negativos = cm.ravel()
    print("Verdaderos positivos: ", v_positivos)
    print("Falsos positivos: ", f_positivos)
    print("Falsos negativos: ", f_negativos)
    print("Verdaderos negativos: ", v_negativos)
    #Métricas
    accuracy = accuracy_score(y_test,y_predicts)
    precision = precision_score(y_test,y_predicts,average='binary')
    recall = recall_score(y_test,y_predicts,average='binary')
    f1 = 2*((recall*precision)/(recall+precision))
    print (accuracy, precision, recall, f1,"\n")
```

4.4.2 Resultados obtenidos en el entrenamiento del modelo. Tomando como base el dataset recolectado que contiene 100.000 tweets de diferentes zonas del país, se realizó el pre procesado y procesado de estos datos. De esta forma se identificaron 1.816 tweets que contienen una o más palabras clave pertenecientes al grupo de síntomas relacionados con depresión y 1.377 de ellos categorizados como negativos por Senpy API.

Con estos datos se procede a realizar el entrenamiento de los algoritmos de clasificación Random Forest Classifier y Multinomial Naive Bayes, y posteriormente la ejecución de los mismos logrando identificar 433 y 378 posibles casos de depresión respectivamente (Ver Tabla 3).

Tabla 3: Resultados obtenidos en entrenamiento de clasificadores.

CONCEPTO	CANTIDAD
Tweets recolectados	100.000
Tweets asociados a un síntoma depresivo a través de filtrado por palabras clave	1.816
Tweets categorizados como negativos por Senpy API	1.377
Tweets identificados como posible caso de depresión por Random Forest	435
Tweets identificados como posible caso de depresión por Multinomial Naive Bayes	367

4.4.2.1 Síntomas encontrados. El conteo de temáticas proporcionadas por el etiquetado octal en la sección 4.1 se realiza definiendo la función contar, en donde un método *for* se encarga de analizar la columna de temática y visualizar su conteo empleando la librería matplotlib (ver Figura 20).

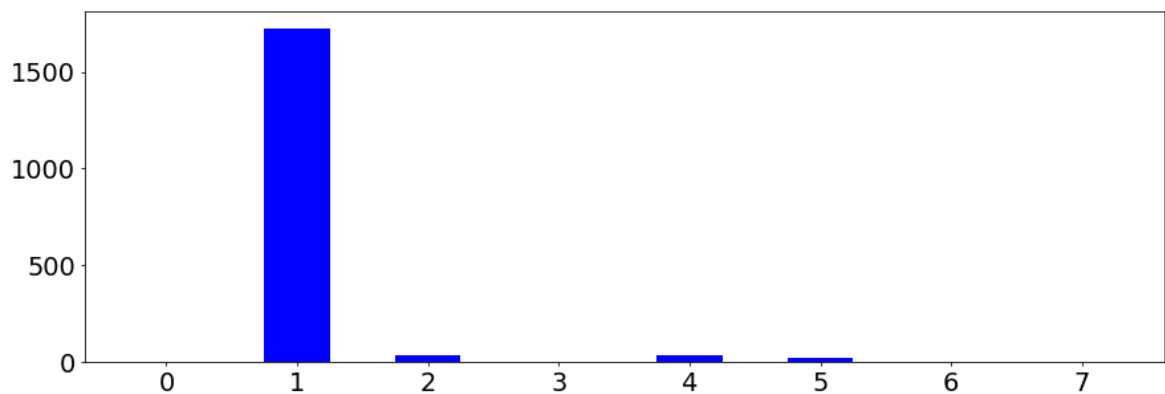
Figura 20: Sección de código - Visualización de temáticas encontradas

```
def contar(serie):
    # 0,1,2,3,4,5,6,7
    eje_y = [0,0,0,0,0,0,0,0]
    for dato in serie:
        if dato == 0:
            eje_y[0] +=1
        elif dato == 1:
            eje_y[1] +=1
        elif dato == 2:
            eje_y[2] +=1
        elif dato == 3:
            eje_y[3] +=1
        elif dato == 4:
            eje_y[4] +=1
        elif dato == 5:
            eje_y[5] +=1
        elif dato == 6:
            eje_y[6] +=1
        elif dato == 7:
            eje_y[7] +=1
    return eje_y

eje_x = [0,1,2,3,4,5,6,7]
eje_y = contar(dataset_global_v['tematica'])
width = 0.5
f, ax = plt.subplots(figsize=(15,5),dpi=70)
ax.tick_params(axis='both', which='major', labelsize=20)
rects1 = plt.bar(eje_x, eje_y, width,
                 color='blue',
                 error_kw=dict(elinewidth=2,ecolor='red'))
#plt.scatter(eje_x,eje_y)
plt.show()
print (eje_y)
```

Por medio del etiquetado octal se lograron identificar cuatro temáticas en particular durante el entrenamiento de los clasificadores por medio de palabras clave. En la *Figura 21* se muestra la etiqueta del síntoma y el número de tweets relacionados a él, en el eje “Y” y “X” correspondientemente.

Figura 21: Síntomas encontrados en el entrenamiento de clasificadores, siendo ‘estado depresivo’ el valor 1, ‘falta de energía’ el valor 2, ‘falta de energía y depresivo’ el valor 3, ‘insomnio’ el valor 4, ‘insomnio y depresivo’ el valor 5, ‘insomnio y falta de energía’ el valor 6 y ‘todas las anteriores’ el valor 7



El primer síntoma correspondiente a “*depresivo*” cuenta con 1.727 tweets y el segundo, “*falta de energía*”, cuenta con 37 trinos. Por su parte las categorías “*insomnio*” e “*insomnio y depresivo*” obtuvieron 33 y 19 tweets cada una. Síntomas como “*falta de energía y depresivo*”, “*insomnio y falta de energía*” y “*todos los anteriores*” no obtuvieron ningún trino relacionado con estas categorías.

4.4.2.2 Análisis de métricas. El análisis de las métricas obtenidas se realiza a través de la matriz de confusión. Para su visualización no se emplea la característica de normalizado, con el fin de mostrar el conteo y no las proporciones. Demás

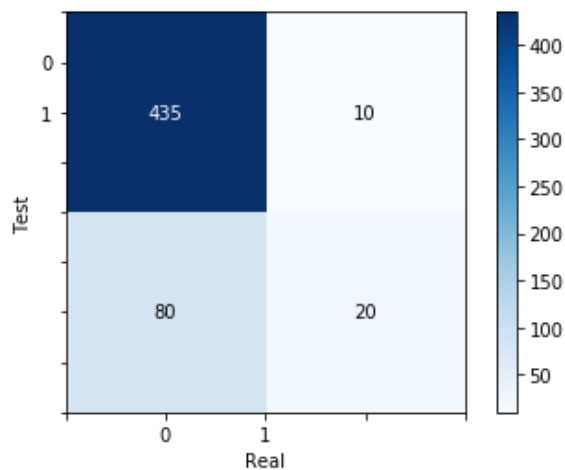
parámetros de visualización son proporcionados por la librería *matplotlib* como se puede observar en la *Figura 22*.

Figura 22: Sección de código - Matriz de confusión

```
# MATRIZ DE CONFUSIÓN
def plot_cm(cm, etiquetas=['0','1'], normalizado=False, cmap=plt.cm.Blues):
    if normalizado:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
    fig = plt.figure()
    ax = fig.add_subplot(111)
    cax = ax.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title('Matrix de confusión del clasificador')
    fig.colorbar(cax)
    ax.set_xticklabels([''] + etiquetas)
    ax.set_yticklabels([''] + etiquetas)
    fmt = '.2f' if normalizado else 'd'
    thresh = cm.max() / 2.
    for i in range(cm.shape[0]):
        for j in range(cm.shape[1]):
            ax.text(j, i, format(cm[i, j], fmt),
                    ha="center", va="center",
                    color="white" if cm[i, j] > thresh else "black")
    plt.ylabel('Test')
    plt.xlabel('Real')
    plt.show()
```

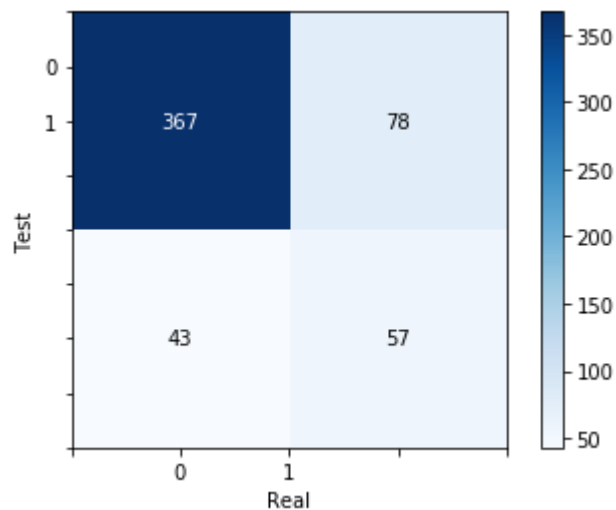
Habiendo obtenido el número de posibles casos detectados por parte de cada clasificador, se realiza una comparación entre los demás resultados que se produjeron durante este proceso y la clasificación manual realizada anteriormente mediante una matriz de confusión.

Figura 23: Matriz de confusión del clasificador Random Forest.



En la *Figura 23* podemos observar que de todas las clasificaciones realizadas por el clasificador Random Forest, 440 pertenecen al grupo de verdaderos positivos, es decir, tweets depresivos predichos como depresivos, además de 80 falsos negativos, 10 falsos positivos y 20 verdaderos negativos. En consecuencia, un clasificador competente a la hora de identificar cuando hay un posible caso de depresión y no tan preciso para descartar los que no.

Figura 24: Matriz de confusión del clasificador Multinomial Naive Bayes.



La matriz de confusión del Multinomial Naive Bayes por su parte, evidencia 367 tweets depresivos clasificados como depresivos, es decir, verdaderos positivos, 43 falsos negativos, 78 falsos positivos y 57 verdaderos negativos (ver Figura 24).

Por último, se efectúa la evaluación de estos algoritmos con las métricas anteriormente mencionadas, con el fin de determinar su eficacia a la hora de realizar el proceso de identificación de tweets como posibles casos de depresión.

Tabla 4: Evaluación de clasificadores

ALGORITMO DE CLASIFICACIÓN	MÉTRICA			
	Accuracy	Precision	Recall	F1 Score
Random Forest Classifier	0.83486	0.66666	0.2	0.30769
Multinomial Naive Bayes	0.77798	0.42222	0.57	0.4851

Como se muestra en la Tabla 4, a pesar de que el clasificador *Random Forest* resulta ser más exacto a la hora de identificar posibles casos de depresión que *Multinomial Naive Bayes*, este último tiene un mejor rendimiento pues hay un mayor balance entre la precisión y la exhaustividad de sus predicciones, teniendo en cuenta las métricas de *recall* y *F1 Score*.

4.4.2.3 Frecuencia de repetición de palabras. Para efectuar el conteo y visualización de la frecuencia de repetición de palabras se crea el conjunto de datos *vocabulario*, en donde llegará el nombre de cada columna creada en la vectorización de los textos. En este caso cada columna corresponderá a una palabra diferente y se agruparán en un solo conjunto gracias la función *vectorizer.get_feature_names*. Tomando los datos del conjunto *cuenta* (ver Figura 13) se procede a realizar el conteo que se guardará mediante un proceso *for* en *frecuencia_n_gramas*. Posteriormente para la visualización se hará uso de la librería *matplotlib* con las propiedades indicadas en la *Figura 25*.

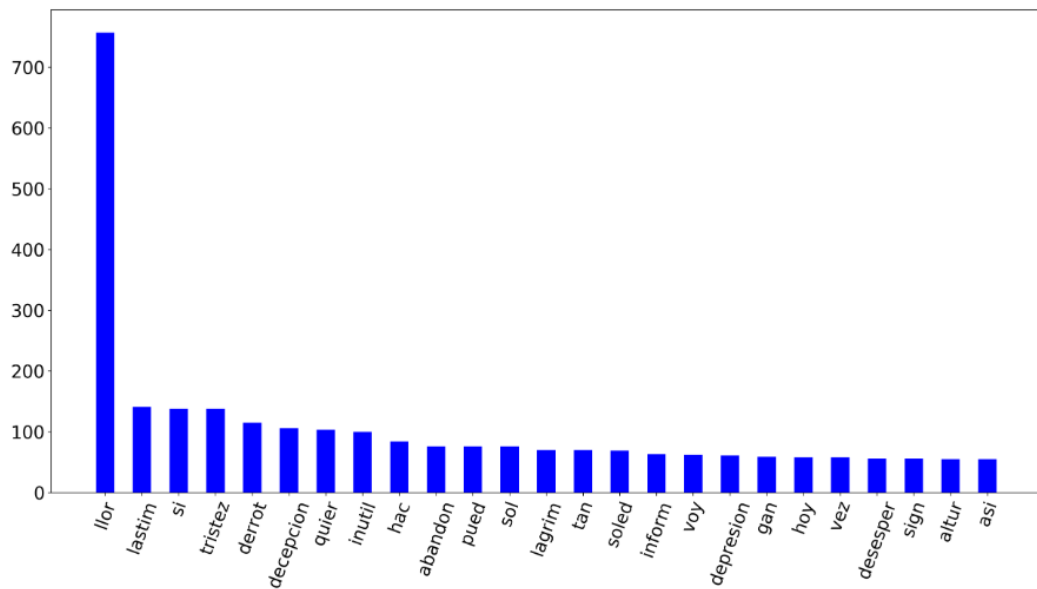
Figura 25: Sección de código - Visualización de frecuencia de repetición

```
# Conteo de características
vocabulario = vectorizer.get_feature_names()
dist = np.sum(cuenta.toarray(), axis=0)
frecuencia_n_gramas = {}
for etiqueta, conteo in zip(vocabulario,dist):
    frecuencia_n_gramas[etiqueta] = conteo

# Organización y visualización de características principales
CANTIDAD = 25
frec_n_grams_organizado = sorted(frecuencia_n_gramas.items(), key=lambda frec_n_grams_organizado: frec_n_grams_organizado[1],
                                reverse=True)
frecs = np.array(frec_n_grams_organizado[:CANTIDAD])
eje_x = frecs[:,0]
eje_y = [int(i) for i in frecs[:,1]]
width = 0.5
f, ax = plt.subplots(figsize=(20,10),dpi=300)
ax.tick_params(axis='both', which='major', labelsize=20)
plt.xticks(rotation=70)
rects1 = plt.bar(eje_x, eje_y, width,
                 color='blue',
                 error_kw=dict(elinewidth=2,ecolor='red'))
plt.scatter(eje_x,eje_y)
plt.show()
```

De esta manera se presenta el listado de palabras con mayor frecuencia de repetición con los datos de entrenamiento:

Figura 26: Frecuencia de repetición de palabras estemizadas con datos de entrenamiento.

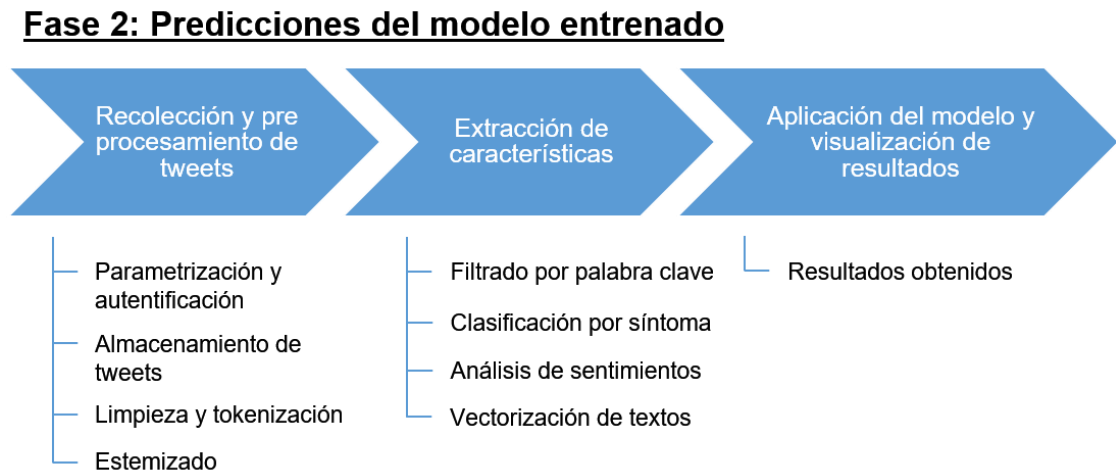


Como se puede observar en la Figura 26, algunas de las palabras con mayor frecuencia de repetición son: llorar, lastimado, tristeza, derrotado, decepción, inútil; pertenecientes a las temáticas mencionadas anteriormente y que están relacionadas con síntomas depresivos.

5. PREDICCIONES DEL MODELO ENTRENADO

En este capítulo se describe en detalle la fase 2: Predicciones del modelo, presentando las etapas y actividades diseñadas y realizadas, así como el código fuente desarrollado. La Figura 27 presenta las actividades para cada una de las etapas de la fase 2.

Figura 27: Etapas de la fase 2 predicciones del modelo



5.1 RECOLECCIÓN Y PRE PROCESAMIENTO DE TWEETS

La recolección de tweets de esta segunda fase se lleva a cabo con la ciudad de Medellín, ciudad con mayor índice de casos de depresión en Colombia de acuerdo con el Boletín de Salud Mental de la subdirección de enfermedades no transmisibles del Ministerio de Salud. Para tal fin, se recolectaron 74.276 tweets desde el 17 de marzo hasta el 17 de mayo de 2020, a los cuales se les aplicaron los procesos de limpieza y tokenización y estemizado, haciendo uso de la clase TLN descrita en las secciones 4.1.3 y 4.1.4.

5.2 EXTRACCIÓN DE CARACTERÍSTICAS

Para la extracción de características en esta segunda fase se realizan los procesos de filtrado por palabra clave y la clasificación por síntoma, donde se encontraron 1.935 publicaciones asociadas a un síntoma depresivo, el análisis de sentimientos a través de Senpy API en el cual 1.377 tweets fueron categorizados como negativos y finalmente la creación de una matriz de densidad propia de este nuevo conjunto de datos. Los procesos anteriores indicados en la extracción de características se llevaron a cabo siguiendo los mismos procedimientos descritos en la sección 4.2.

5.3 APLICACIÓN DEL MODELO Y VISUALIZACIÓN DE RESULTADOS

El modelo entrenado fue aplicado a través de una tubería de datos y su clasificación realizada por los dos algoritmos se muestra a continuación en la Tabla 5:

Tabla 5: Resultados obtenidos para la ciudad de Medellín

CONCEPTO	CANTIDAD
Tweets recolectados	74.276
Tweets asociados a un síntoma depresivo a través de filtrado por palabras clave	1.935
Tweets categorizados como negativos por Senpy API	1.377
Tweets identificados como posible caso de depresión por Random Forest	527
Tweets identificados como posible caso de depresión por Multinomial Naive Bayes	465

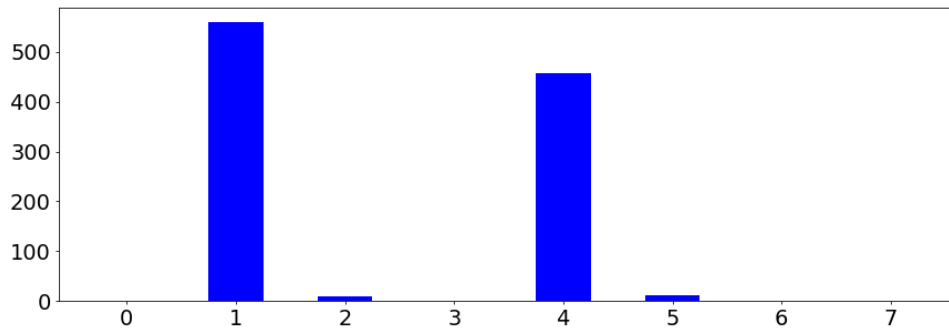
Para agrupar los resultados obtenidos por ambos clasificadores identificando posibles casos de depresión, se realiza la unión de los mismos descartando aquellos trinos que fueron categorizados como no depresivos por ambos clasificadores. Por tal motivo se emplea una suma en álgebra booleana de ambos conjuntos de datos, bajo la cual un tweet será depresivo si fue identificado por uno de los dos clasificadores o por ambos (Ver Figura 28). De esta manera se detecta un total de 1.041 posibles casos predichos para la ciudad de Medellín.

Figura 28: Sección de código - Casos identificados en la ciudad de Medellín

```
# Eliminación de contenido clasificado como NO depresivo por Los dos modelos
i = 0
lista_negativos = []
for (p1, p2) in zip(CLASIFICADORES['RaF']['y_predicts'], CLASIFICADORES['MNB']['y_predicts']):
    if (p1+p2) == 0:
        lista_negativos.append(i)
        i += 1
m_resultados = dataset_procesado.drop(lista_negativos)
m_resultados = m_resultados.drop(['pred_RaF', 'pred_MNB'], axis=1)
```

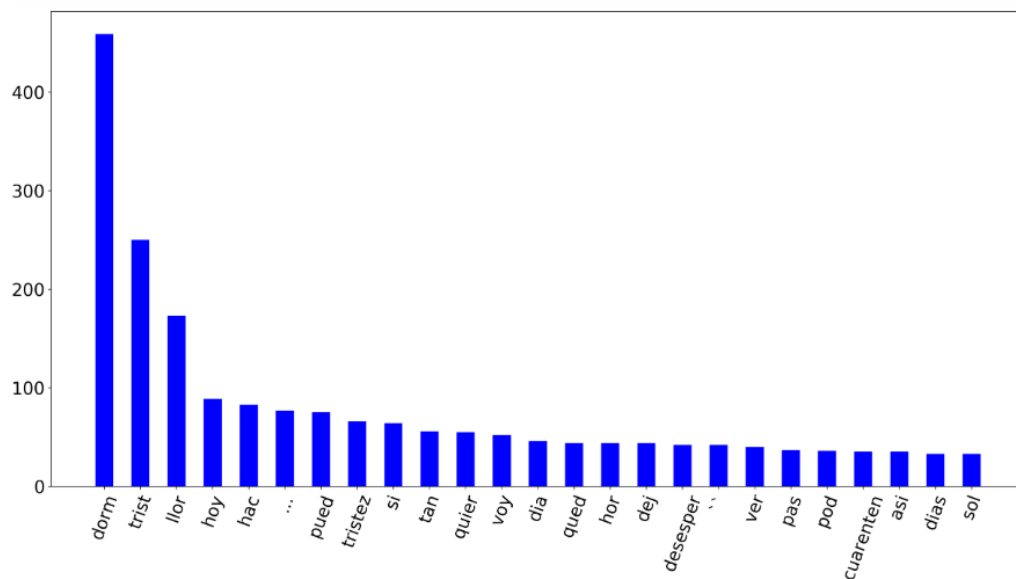
Adicionalmente dentro de los síntomas identificados, como se puede observar en la Figura 29 presentada en la siguiente página, se destacan el estado “*depresivo*” con 561 resultados, “*falta de energía*” con 9 resultados, “*falta de energía y depresivo*” con 1 resultado, “*insomnio*” con 458, “*insomnio y depresivo*” con 11 resultados y finalmente la categoría “*falta de energía e insomnio*” con 1.

Figura 29: Síntomas encontrados en la ciudad de Medellín, siendo ‘estado depresivo’ el valor 1, ‘falta de energía’ el valor 2, ‘falta de energía y depresivo’ el valor 3, ‘insomnio’ el valor 4, ‘insomnio y depresivo’ el valor 5, ‘insomnio y falta de energía’ el valor 6 y ‘todas las anteriores’ el valor 7



Para concluir con los resultados obtenidos aplicando el modelo en la ciudad de Medellín, se realiza el conteo y visualización de frecuencia de repetición de palabras, en el cual se destacan palabras como dormir, triste, llorar y puedo, entre otras (ver Figura 30).

Figura 30: Palabras estemizadas con mayor frecuencia en la ciudad de Medellín



6. DIVULGACIÓN Y CUMPLIMIENTO DE OBJETIVOS

6.1 PARTICIPACIÓN EVENTO ACADÉMICO

Los resultados preliminares del presente trabajo de grado fueron presentados en el evento “TICs, Big Data e Investigación en Salud”, organizado por el Departamento de Salud Pública de la Facultad de Salud UIS (12 de agosto de 2019, auditorio Leonardo Amaya), junto con los panelistas Rafael Ortiz de la Foscal y Álvaro Osorio de la Universidad de Alberta en Canadá.

Figura 31: Participación en evento académico.



6.2 PROPUESTA DE INVESTIGACIÓN

El presente trabajo de grado fue la base de la propuesta de investigación titulada “*Librería software basada en aprendizaje automático y deep learning para el apoyo a la detección de depresión y prevención del suicidio mediante el análisis de publicaciones en redes sociales*”, presentada a la convocatoria de la Vicerrectoría de Investigación y Extensión UIS 2020, la cual está en estudio actualmente. La aprobación de este proyecto permitirá fortalecer la línea de investigación del procesamiento del lenguaje natural y aprendizaje automático, y contribuir a la detección temprana la depresión y consecuente prevención del suicidio en ciudades de Colombia.

6.3 CUMPLIMIENTO DE OBJETIVOS

La tabla presentada en la siguiente página muestra cada uno de los objetivos específicos del presente proyecto y el avance alcanzado para cada uno de ellos.

Tabla 6: Cumplimiento de objetivos

OBJETIVO GENERAL		
<p>Desarrollar un prototipo software que permita, mediante técnicas de aprendizaje automático (machine learning), clasificar publicaciones provenientes de la red social Twitter para así identificar tendencias de depresión en Colombia.</p>		
OBJETIVO ESPECÍFICO	CUMPLIMIENTO	EVIDENCIA.
Diseñar y programar las funcionalidades correspondientes al descargue y pre-procesamiento de los trinos.	100%	Ver sección 4.1.
Diseñar y programar un analizador de texto que identifique las temáticas y asigne la polaridad a los textos pre-procesados.	100%	Ver sección 4.2.
Seleccionar dos algoritmos de aprendizaje automático a utilizar en el prototipo, de los más usados en el área del procesamiento de lenguaje natural, a saber: Naive Bayes, Random Forest, Regresión Logística, Support Vector Machine.	100%	Ver sección 4.3.
Realizar la clasificación de los textos mediante los dos algoritmos seleccionados y evaluar los correspondientes resultados.	100%	Ver sección 4.4.
Diseñar y programar la visualización de resultados.	100%	Ver sección 4.4.

7. CONCLUSIONES Y TRABAJO FUTURO

En el presente proyecto se desarrolló un prototipo software que busca identificar tendencias de depresión en Colombia, mediante técnicas de aprendizaje automático en la red Social Twitter. Las siguientes son las principales conclusiones del trabajo realizado.

- El prototipo desarrollado permitió evidenciar que el procesamiento del lenguaje natural y los algoritmos de aprendizaje automático pueden ayudar a identificar y predecir posibles casos de depresión en Colombia.
- Una de las principales dificultades para la clasificación de los mensajes fue el manejo de las negaciones, intensificadores y ambigüedades, oraciones fuera de contexto y figuras como el sarcasmo y la ironía.
- Se evidenció que el algoritmo de clasificación Multinomial Naive Bayes tiene un mejor rendimiento a la hora de clasificar las publicaciones provenientes de la red social Twitter.
- El lenguaje Python demostró tener una gran potencialidad y facilidad para trabajar con problemas relacionados con tratamiento de texto y aprendizaje automático.
- Personalmente fue satisfactorio trabajar en el campo del procesamiento del lenguaje natural, contribuyendo en el ámbito de la salud mental en Colombia.

7.1 TRABAJO FUTURO

Los siguientes aspectos se consideran luego de haber realizado el presente proyecto, como posibles mejoras a realizar:

- **Seguimiento a posibles casos detectados:** Realizar un análisis y seguimiento de la línea de tiempo de las publicaciones de los usuarios identificados, con el fin de evidenciar la recurrencia de temáticas depresivas.
- **Análisis pragmático de los textos con especialista en área de la salud:** Incorporar el contexto de los tweets dentro del análisis en conjunto con un experto en salud mental.
- **Incrementar el tamaño del dataset:** Recolectar una mayor cantidad de datos a nivel de ciudad para detectar tendencias significativas y replicar el proceso con las demás ciudades.
- **Clasificación demográfica:** Añadir atributos demográficos a la clasificación para catalogar edades, género y geolocalización.
- **Incluir otros algoritmos de clasificación:** Analizar el rendimiento de clasificadores diferentes a los escogidos en el presente trabajo.
- Se recomienda fortalecer la línea de procesamiento del lenguaje natural en la escuela de Ingeniería de Sistemas, dado su auge y sus aplicaciones.

BIBLIOGRAFÍA

CHANCELLOR, Stevie; MITRA, Tanushree y DE CHOUDHURY, Munmun. 2016. Recovery Amid Pro-Anorexia. En: CHI Conference on Human Factors in Computing Systems - CHI '16, 2111p, New York, New York, USA. ACM Press.

CONWAY Mike y O'CONNOR Daniel. 2016. Social Media, Big Data, and Mental Health: Current Advances and Ethical Implications. *Current Opinion in Psychology*, 9:77–82.

DE CHOUDHURY, Munmun, *et al.* 2016. Discovering Shifts to Suicidal Ideation from Mental Health Content in Social Media. In the 2016 CHI Conference on Human Factors in Computing Systems, 2098p, San Jose, CA, USA. ACM Press.

DEVELOPER. Publish and manage Tweets, and analyze Tweets data. [En línea]. [Consultado: 18 de mayo de 2020]. Disponible en: <https://developer.twitter.com/en/products/tweets>

GHANDI, Rohith. Naïve Bayes Classifier [En línea]. Towards data science, 2018. [Consultado: 8 de mayo de 2020]. Disponible en: <https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c>

JUPYTER. The jupyter notebook [En línea]. Project Jupyter, (16 de mayo de 2020). [Recuperado: 18 de mayo de 2020]. Disponible en: <https://jupyter.org/>

MACLEAN, Diana, *et al.* 2015. An Analysis of an Online Health Forum Dedicated to Addiction Recovery. En: 18th ACM Conference on Computer Supported Cooperative Work & Social Computing, CSCW '15, 1511p, New York, NY, USA. ACM.

MOWERY, Danielle, *et al.* 2016. Towards Automatically Classifying Depressive Symptoms from Twitter Data for Population Health. [En línea]. En: workshop on computational modeling of people's opinions, personality, and emotions in social media. Osaka, Japón: The Coling, 191p. (Recuperado en agosto de 2019). Disponible en: <https://www.aclweb.org/anthology/W16-43.pdf>

MYSLÍN, Mark, *et al.* 2013. Using Twitter to Examine Smoking Behavior and Perceptions of Emerging Tobacco Products. *Journal of Medical Internet Research*, 15(8):e174.

NADEEM, Moin, *et al.* Identifying Depression on Twitter. [En línea]. (Recuperado en agosto de 2019). Disponible en: <https://arxiv.org/ftp/arxiv/papers/1607/1607.07384.pdf>

NLTK. NLTK 3.5 Documentation [En línea]. [Consultado: 18 de mayo de 2020]. Disponible en: <https://www.nltk.org/>

NUMPY. The fundamental package for scientific computing with Python [En línea]. [Consultado: 18 de mayo de 2020]. Disponible en: <http://www.numpy.org/>

ORGANIZACIÓN MUNDIAL DE LA SALUD: Depresión [En línea]. [Consultado: 1 de agosto de 2019] Disponible en: <https://www.who.int/topics/depression/es/>

PANDAS. Pandas [En línea]. [Consultado: 18 de mayo de 2020]. Disponible en: <https://pandas.pydata.org/>

PYTHON. [Consultado: 18 de mayo de 2020]. Disponible en: <https://docs.python.org/>

SCIKIT-LEARN. Machine learning in Python [En línea]. [Consultado: 18 de mayo de 2020] Disponible en: <https://scikit-learn.org/>

SENPY. Welcome to Senpy's documentation [En línea]. [Consultado: 18 de mayo de 2020]. Disponible en: <https://senpy.readthedocs.io/>

SOBRINO, José. Análisis de Sentimientos en Twitter. [En línea]. Máster Universitario en Ingeniería Informática. Barcelona, España. Universitat Oberta de Catalunya, 2018. [Consultado: 10 de marzo de 2020]. Disponible en: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/81435/6/jsobrinostFM0618memoria.pdf>

SUÁREZ, Daniel. Diseño y desarrollo de un sistema de caracterización de trastornos del sueño mediante Social Media Mining [En línea]. Madrid, España. Universidad Politécnica de Madrid, 2018. [Consultado: 10 de marzo de 2020]. Disponible en: http://vps161.cesvima.upm.es/administrator/components/com_jresearch/files/publications/tfg-dani.pdf

TAMERSOY, Acar; DE CHOUDHURY, Munmun y CHAU, Duen Horng. 2015. Characterizing Smoking and Drinking Abstinence from Social Media. En: 26th ACM Conference on Hypertext & Social Media, 139p, Guzelyurt, TRNC, Cyprus, september 1-4th 2015. ACM.

TWEEPY. Tweepy documentation [En línea]. [Consultado: 18 de mayo de 2020]. Disponible en: <http://docs.tweepy.org>

WIKIPEDIA, THE FREE ENCYCLOPEDIA. Modelo de prototipos [En línea]. [Consultado: 18 de mayo de 2020]. Disponible en: https://es.wikipedia.org/wiki/Modelo_de_prototipos

ZHOU, Victor. A simple explanation of the bag of words model [En línea]. Towards data science, 2019. [Recuperado: 8 de mayo de 2020]. Disponible en: <https://towardsdatascience.com/a-simple-explanation-of-the-bag-of-words-model-b88fc4f4971>