



**FACULTAD DE INGENIERÍA, ARQUITECTURA Y  
URBANISMO**

**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**

**TESIS**

**COMPARACIÓN DE TÉCNICAS DE  
CLASIFICACIÓN DE APRENDIZAJE DE MÁQUINA  
EN EL DIAGNÓSTICO DEL TRASTORNO  
DEPRESIVO LEVE**

**PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO  
DE SISTEMAS**

**Autor(a) (es):**

**Bach. Lucero Cieza Robinson.**

**ORCID: <https://orcid.org/0000-0003-3858-7719>**

**Asesor(a):**

**Mg. Atalaya Urrutia Carlos William**

**ORCID: <https://orcid.org/0000-0002-2761-4868>**

**Línea de Investigación:**

**Infraestructura, Tecnología y Medio Ambiente**

**Pimentel – Perú 2022**

**APROBACIÓN DEL JURADO**  
**COMPARACIÓN DE TÉCNICAS DE CLASIFICACIÓN DE APRENDIZAJE**  
**DE MÁQUINA EN EL DIAGNÓSTICO DEL TRASTORNO DEPRESIVO**  
**LEVE.**

---

**Bach. Lucero Cieza Robinson**  
**Autor**

---

**Mg. Atalaya Urrutia Carlos William**  
**Asesor**

---

**Dr. Vasquez Leyva Oliver**  
**Presidente de Jurado**

---

**Mg., Sialer Rivera María Noelia**  
**Secretario de Jurado**

---

**Mg. Díaz Vidarte Miguel Orlando**  
**Vocal de Jurado**

## **Dedicatorias**

A mi familia, ellos que siempre están en todo momento brindándome el impulso para seguir adelante y lograr el objetivo más anhelado.

A mis abuelos que se encuentran en el cielo y que desde allí me cuidan y guían por el sendero del bien, practicando los valores y enseñanzas que desde niño me inculcaron.

A mis amigos, que siempre estuvieron allí, compartiendo experiencias, anécdotas y enseñanzas, a ellos que son parte de mi formación y que me ayudan a crecer en el ámbito profesional, personal y social.

## **Agradecimientos**

Agradecer a Dios padre todo poderoso, por brindarnos la sabiduría, salud, y la armonía familiar que fueron los principales ejes de motivación para la realización de la presente investigación.

Así mismo un especial agradecimiento a todos los docentes de la Universidad por inculcarnos el conocimiento durante la formación profesional.

Un especial agradecimiento al grupo de Psicólogos de diferentes establecimientos de salud de la provincia de Jaén, toda vez que han dedicado su tiempo para orientar y describir el problema de trastorno mental depresivo en un nivel leve, así como detallar el flujograma para el diagnóstico del mismo.

## **Resumen**

A nivel mundial la depresión lo padece unos 350 millones de seres humanos y el 5% es a nivel de Latinoamérica, es así que, cada veintidós minutos un ser humano intenta hacerse daño, las edades con mayores problemas depresivos son los adolescentes el cual representa el 10%, el 6% adultos mayores de 18 años y 3.5% en niños de 6 a 10 años, en el Perú, el 80% de suicidios es a causa de la depresión, hay un millón setecientos mil personas que presentan cuadro depresivo, pero solo es atendido un 25% con atención especializada y el 65% simplemente no busca ayuda, estudios han demostrado que a nivel del ministerio de salud, el documento técnico llamado "auto escala de Zung", es el más adecuado para la identificación de este problema analizando la medición de la depresión a través de información de aspectos cognitivos, afectivos y somáticos del paciente, dicho documento tiene una especificidad del 63% y sensibilidad del 97%, aprobando un acierto del 82% para discriminar la depresión. En esta investigación se construyó un método que inicia con la elaboración de un dataset de acuerdo a las variables de ingreso y salida así como el nivel de prioridad basados en el cuestionario de Zung, después se realizó la elección de las técnicas de aprendizaje de máquina, utilizadas para tratar casos de diagnóstico de depresión con mayor precisión, entre ellas lograron destacar, naive bayes, árbol de decisión, redes neuronales y máquinas de soporte vectorial, acto seguido se implementó las técnicas mencionadas para ser comparadas y evaluadas según su desempeño, para el desarrollo de las mismas se utilizó la plataforma de google colab con el lenguaje de programación python, según el método propuesto, desarrollado y evaluado se concluye que las redes neuronales tienen una precisión del 100% para el diagnóstico de depresión.

### **Palabras Clave:**

Aprendizaje automático, trastornos depresivos, diagnóstico de depresión, redes neuronales, máquinas de soporte vectorial, árbol de decisión, naive bayes.

## **Abstract**

Globally, depression is suffered by 350 million human beings and 5% is at the level of Latin America, so, every twenty-two minutes a human being tries to harm himself, the ages with the greatest depressive problems are adolescents which represents 10%, 6% adults over 18 years and 3. In Peru, 80% of suicides are caused by depression, there are one million seven hundred thousand people with depressive symptoms, but only 25% are attended with specialized care and 65% simply do not seek help, studies have shown that at the level of the Ministry of Health, the technical document called "Zung self scale", This document has a specificity of 63% and a sensitivity of 97%, approving an accuracy of 82% to discriminate depression. In this research a method was built that starts with the elaboration of a dataset according to the input and output variables as well as the level of priority based on the Zung questionnaire, then the choice of machine learning techniques was made, used to treat cases of depression diagnosis with greater accuracy, among them managed to highlight, naive bayes, decision tree, Neural networks and support vector machines, then the mentioned techniques were implemented to be compared and evaluated according to their performance, for the development of these techniques the google collaborative platform was used with the python programming language, according to the proposed method, developed and evaluated, it is concluded that neural networks have an accuracy of 100% for the diagnosis of depression.

### **Keywords:**

Machine learning, depressive disorders, depression diagnosis, neural networks, support machines, decision tree, naive bayes.

## Índice

<b>I. INTRODUCCIÓN</b> .....	9
<b>1.1. Realidad Problemática</b> .....	9
<b>1.2. Trabajos previos</b> .....	12
<b>1.3. Teorías relacionadas al tema</b> .....	25
<b>1.4. Formulación del Problema</b> .....	44
<b>1.5. Justificación e importancia del estudio</b> .....	44
<b>1.6. Hipótesis</b> .....	45
<b>1.7. Objetivos</b> .....	45
<b>1.7.1. Objetivo general</b> .....	45
<b>1.7.2. Objetivos específicos</b> .....	45
<b>II. MATERIAL Y MÉTODO</b> .....	45
<b>2.1. Tipo y Diseño de Investigación</b> .....	45
<b>2.2. Variables, Operacionalización</b> .....	45
<b>2.3. Población y muestra</b> .....	46
<b>2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad</b> .....	47
<b>2.5. Procedimiento de análisis de datos</b> .....	47
<b>2.6. Criterios éticos</b> .....	48
<b>2.7. Criterios de Rigor Científico</b> .....	49
<b>III. RESULTADOS</b> .....	50
<b>3.1. Resultados en Tablas y Figuras</b> .....	50
<b>3.2. Discusión de resultados</b> .....	57
<b>3.3. Aporte práctico</b> .....	58
<b>IV. CONCLUSIONES Y RECOMENDACIONES</b> .....	77
<b>4.1. Conclusiones</b> .....	77
<b>4.2. Recomendaciones</b> .....	78

<b>REFERENCIAS.....</b>	<b>79</b>
<b>ANEXOS.....</b>	<b>86</b>

## I. INTRODUCCIÓN

### 1.1. Realidad Problemática.

La depresión es un trastorno mental caracterizado por presentar síntomas como: No querer vivir cada día, tristeza, fatiga, etc. esto empobrece el estilo de vida y afecta al entorno familiar y social en todos los escenarios (García & Pérez, 2013).

Según Jiang, y otros, (2020), exponen que esta enfermedad es uno de los trastornos más comunes a nivel mundial y que existe unos 350 millones de humanos, dicho trastorno de salud mental puede ser un serio problema si es de larga duración.

De acuerdo a la intensidad de síntomas los trastornos mentales pueden ser clasificados como leves, moderados o graves (Organización Mundial de la Salud, 2020).

En la figura muestra que la depresión es la causa de suicidios en un 60% y 80%, asimismo nos menciona que de cada 22 minutos un ser humano intenta hacerse daño, las edades con mayores problemas depresivos son los adolescentes el cual representa el 10% entre las edades de 11 y 18 años, el 6% adultos mayores de 18 años y 3.5% en niños de 6 a 10 años. (Instituto Nacional de Salud Mental, 2016).

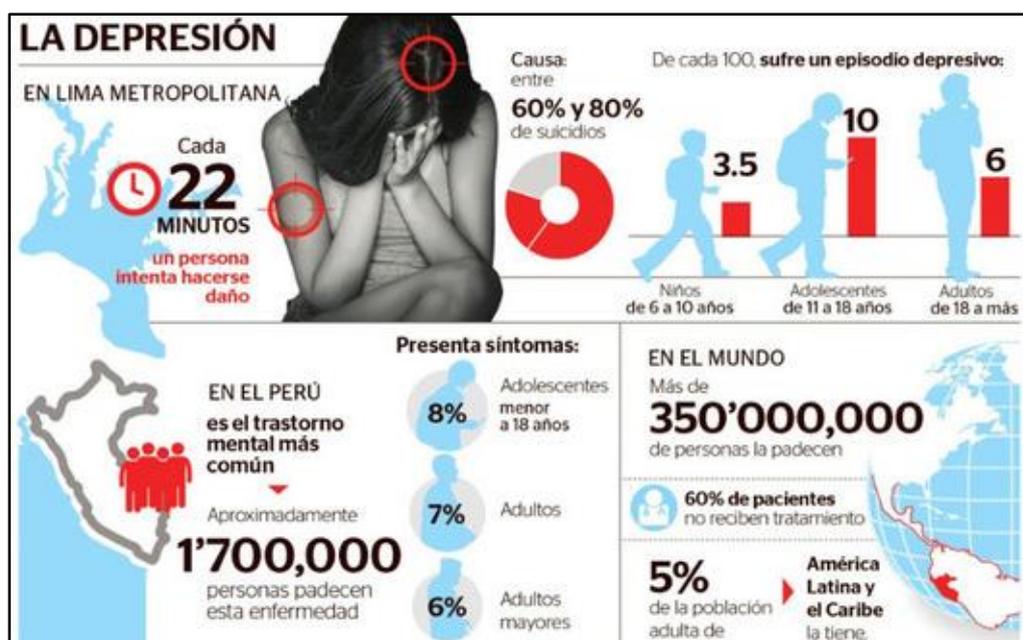


Figura 1. Estadísticas de la Depresión. Fuente: (Instituto Nacional de Salud Mental, 2016)

Según la imagen, en el Perú, el 80% de suicidios es a causa de la depresión, hay un millón setecientos mil personas que presentan cuadro depresivo, pero el solo es atendido un 25% con atención especializada y el 65% simplemente no busca ayuda. (Organizacion Mundial de la Salud, 2020)



Figura 2. Reporte Estadístico de la Depresión. Fuente: (Organizacion Mundial de la Salud, 2020)

El Psiquiatra (Freddy Vásquez, 2020) presidente de la Sociedad Peruana de Prevención del Suicidio, comenta que existe un alto porcentaje de suicidios en el sexo masculino, alertándose más en adolescentes de 12 a 14 años de edad.

Las señales de alerta frente a este caso son de gran importancia, ya que ayudará a identificar los síntomas y signos depresivos, es importante solicitar el apoyo especializado psicológico para su tratamiento y monitoreo, los más destacados síntomas que se presentan en la depresión son, la alteración del sueño, tristeza profunda, llantos insaciables, ganas de llorar sin razón alguna, desesperanza, desinterés en la vida cotidiana, cansancio, fatiga, aislamiento de amistades y familia, en lo general se mantienen encerrados en su cuarto (Tuñoque, 2020).



Figura 3. Síntomas de Depresión. Fuente: (Tuñoque, 2020).

Según (Yalamanchili, y otros, 2020), indican que para el diagnóstico de dicha enfermedad pasan por una serie de entrevistas con la familia y el paciente, desarrollando cuestionarios, análisis clínicos, entre otros. Lo cual conlleva a que se requiera profesionales médicos con una formación idónea. En la era actual del machine learning, aporta algoritmos para la automatización de todas las tareas detectando la depresión de una manera más fácil y precisa, considerando que este debe contar con utilizar menos recursos y proporcionar resultados óptimos sobre el tema.

El objetivo del presente trabajo es realizar un estudio comparativo de algoritmos de clasificación para el diagnóstico del trastorno depresivo leve dentro del campo de la psicología, resaltando indicadores de exactitud, precisión, tiempo de respuesta, consumo de memoria ram y consumo de CPU.

Con la investigación tratamos de optimizar los procedimientos plasmados en la última guía de diagnóstico y tratamiento de la depresión que fue aprobada por el Ministerio de Salud en el año del 2008, dicho documento técnico solo está enfocada al tratamiento farmacológico dejando de lado otros puntos de gran interés, sin embargo, en la mayoría de los hospitales y establecimientos de salud urge acceder a una atención adecuada. A nivel nacional de manera aproximada el 90% de los pacientes tamizados no reciben un tratamiento oportuno por las limitaciones en el diagnóstico, otro de las razones es el escaso

número de médicos especialistas (psiquiatra) que solo están concentrados en las principales ciudades y no en el primer nivel de atención (Bellido Riquelme , Aragonés Alosilla, Bellido Riquelme, Castro Paredes, & Ramírez Freyre, 2008)

## **1.2. Trabajos previos.**

Para (Bhanusree , Nikhil , Maruthi Saketh , & Sandeep , 2020), en su publicación “*Real-time Acoustic based Depression Detection using Machine Learning Techniques*”, estudian el problema de la depresión a través de características acústicas con el objetivo de realizar un modelo de clasificación y verificar si una persona está o no con depresión a través del habla utilizando ciertas características, el método propuesto de esta investigación inicia en con los cuestionarios tipo entrevistas, para ello el apoyo de un psiquiatra fue fundamental por la serie pruebas mentales que ayudan a confirmar el estado y clasificarlos de manera objetiva. Posteriormente se extraen varias características acústicas obteniéndose mediante la base de datos DIACWOZ, resultado del taller de reconocimiento de emociones, depresión y estados de ánimo durante el año 2016, luego se extraen los rasgos a través de fotogramas de un repositorio en análisis de voz, obteniendo 189 pacientes de los cuales 56 tienen depresión y 133 no, en la base de datos se evidencia que existe 60% para entrenamiento, 15% para desarrollo y 25 % para pruebas, se utilizó el algoritmo máquinas de vectores de soporte (SVM) teniendo una precisión del 95%, en seguida esta los bosques aleatorios con un 47% y regresión logística con un 79%, queda demostrado que el SVM tiene una precisión aprobada para el diagnóstico de la depresión.

Según los autores (Vaishali Narayanrao & Lalitha Surya Kumari 2020) abordan el tema de la depresión en su artículo “*Analysis of Machine Learning Algorithms for Predicting Depression*”, En el mencionado documento enfrenta un problema de predecir la depresión.

El método que los autores propusieron es basado en la tecnología de inteligencia artificial con algoritmos de clasificación para identificar si una persona está o no deprimida.

Para el presente caso los autores elaboraron un data set utilizando palabras clave de tweets de la red social de Twitter, donde utilizaron aprendizaje automático para obtener un resultado limpio basado en reglas, Los resultados de los algoritmos de agrupación en clústeres, de K-mens, SVM, KNN y Conjunto de árboles dan un resultado del 90% de precisión para las validaciones de las etiquetas registradas.

La depresión son unas de las principales causas del suicidio, el mismo que se evita si existe una detección de esta enfermedad en una etapa temprana, haciendo uso de un mecanismo de aprendizaje profundo para la identificación de la depresión.

Los autores (Asad, Mahmud Pranto, Afreen, & Islam, 2019), nos explican en su investigación *“Depression Detection by Analyzing Social Media Posts of User”*, desarrollado en Dhaka, sobre el problema de la depresión utilizando información de publicaciones en las redes sociales, el método de estudio inicia con la exploración de publicaciones en twitter utilizando la biblioteca Beautiful Soup, que sirve para el análisis de datos en documentos HTML, para ello es importante identificar al usuario en twitter y desde la línea de comandos se envía una solicitud, para extraer textos que se encuentren asociada a una etiqueta <p>, luego la información obtenida se guarda en formato json, para los usuarios de facebook se pidió envíen sus publicaciones en el mismo formato, posteriormente ambos archivos fueron convertidos de json a csv, quedando listo para ingresar al modelo de aprendizaje automático. Acto seguido se inició con el procesamiento de datos utilizando la biblioteca NLTK y tokenización, esto con la finalidad de tener frases que realmente necesitamos eliminando todas las palabras de parada y los retweets, construyéndose dos bucles, el externo que selecciona cada fila y el interno que tokeniza almacenándose en una matriz numpy. En este estudio se emplearon dos clasificadores uno de ellos máquinas de vectores de soporte (SVM) y naive bayes (NB), para la aplicación de ambos algoritmos se separó dos secciones principales, uno de prueba y otro de entrenamiento, para ello se utilizó el módulo “train\_test\_split”, de la librería de sklearn que permite la división en sub

grupos, luego estos se ajustaron al modelo de NB, que es el encargado de ejecutar la predicción determinando tres criterios para evaluar el nivel de la depresión, de 1 a 25% normal, 26 a 40% leve, 41 a 55% límite, 56 a 70% moderada, 71 a 85% severa y de 85 a 100% extrema. Se procesaron 100 tweets de los cuales el 35% se clasificó como no deprimido y el 65% con depresión, por otro lado, se procesó a 50 personas en facebook donde se obtuvo 38% con depresión y el 62% como no deprimidos. Basado en los resultados se puede confirmar que el algoritmo NB obtuvo el 100% de precisión para predecir pacientes con depresión dentro del contexto de las redes sociales.

Los investigadores (Gerych, Agu, & Rundensteiner, 2019), estudian el problema de la depresión en su artículo *“Classifying Depression in Imbalanced Datasets Using an Autoencoder- Based Anomaly Detection Approach”*, desarrollado en USA, los autores mencionan que los métodos tradicionales de diagnóstico de depresión son cuestionarios médicos donde el paciente responde teniendo un resultado subjetivo y que sería deseable adoptar nuevos métodos de aprendizaje automático para su detección de manera autónoma, para este estudio desarrollaron un método para mitigar características de clases en la detección de esta enfermedad, el desarrollo del método está dividido en varias etapas, en primer lugar se utiliza autocodificadores que es un tipo de red neuronal que se entrena en base a datos de entrada y salida para minimizar los errores entre estos, proyectando características de localización GPS, de usuarios no deprimidos, luego el autocodificador entrenado realiza la clasificación agrupándolos como deprimidos y no deprimidos, utilizando el algoritmo máquinas de vectores de soporte (SVM), realiza la clasificación y posterior evaluación del modelo logrando un 92% de exactitud, así mismo los autores recomiendan utilizar otras características de movilidad para identificar la depresión.

Los autores (Malaquias, y otros, 2019), nos explican en su artículo, *“Classification and characterization of children and adolescents with depressive symptomatology using machine learning”*, Estudian el problema

de la depresión y su complejidad en factores biológicos, sociales y psicológicos, identificando inconvenientes en el pronóstico y diagnóstico. Frente a ello, un gran aliado son las técnicas de aprendizaje automático, utilizadas con mayor frecuencia para caracterizar o clasificar diferentes enfermedades, entre ellas la depresión mayor en adolescentes y niños de Brasil, para el desarrollo del estudio se propone un método que inicia con la elaboración de una base de datos con información de adolescentes y niños de 10 a 16 años, identificándose 75 atributos en 219 mujeres y 158 hombres, además de 377 indicadores de síntomas depresivos clasificados por alta y baja, por otro lado se consideraron los atributos más importantes que están vinculados al sexo, edad, tiempo que pasan con sus padres, si consume medicamentos, puntuaciones de auto informes (CDI, YSR), grado de instrucción de los padres, tratamiento psiquiatra, quien vive con el niño o adolescente, tiene problemas de ansiedad, problemas de conducta y otras preguntas importantes consideradas con la comunidad de salud mental. Luego de ello se inicia con la etapa de pre procesamiento de la información optando por estrategias sencillas antes de la aplicación de los algoritmos, secuencialmente se procedió al procesamiento que constaba en la normalización de datos que faltan, la eliminación de atributos con información escasa, consistencia de datos, codificación de numérica a nominal, transformación de atributos a binario, discretización de atributos, identificación de atributos de clasificación, equilibrio de la base de datos, separación de conjuntos de datos de entrenamiento y de prueba, posteriormente se empezó a trabajar con los algoritmos que fueron elegidos por su aceptación en el tema de depresión, entre ellos están el algoritmo J48 que está basado en la implementación de árbol de decisión C4.5 en el entorno de Weka, también se seleccionó el algoritmo SimpleCart que hace una partición binaria recursiva en la elaboración del árbol de clasificación y regresión, para el algoritmo Multilayer perceptron se definió dos capas ocultas la primera con 5 neuronas y la segunda con 3, en el entrenamiento de SVM fue necesario el ajuste de 34 parámetros principales, la fusión del kernel y la gama, para lograr los resultados esperados, de todo lo mencionado se puede concluir que SVM es el algoritmo con mejor precisión

con un 90% y 87.3% para características depresivas altas y bajas respectivamente. Por otro lado, los autores recomiendan que en trabajos futuros se aplique modelos en una base de datos que utilice otros métodos de clasificación en niños y adolescentes.

Los autores (Khalil & Al-Jumaily, 2017), en su artículo "*Machine learning based prediction of depression among type 2 diabetic patients*", nos explican el problema del diagnóstico de la depresión en pacientes que padecen enfermedad de diabetes de tipo 2, este estudio fue realizado en la universidad de tecnología de Sídney en el país de Australia, los investigadores afirman que las técnicas de aprendizaje automático pueden ayudar a construir un modelo compacto para la asignación de etiquetas basándose en todas las características que el paciente presente, el método está basado en primer lugar en cargar el conjunto de datos de entrada, que está compuesto por 39 atributos los cuales fueron obtenidos mediante entrevistas con los pacientes, luego de la recolección de datos se pasó a seleccionar cuatro técnicas ideales para demostrar resultados precisos, entre ellos se encuentran las máquinas de vectores de soporte (SVM), K vecinos más cercanos, Fuzzy c-Means y redes neuronales probabilísticas, luego se procedió a entrenar el conjunto de datos con los clasificadores seleccionados con la finalidad de calcular la medida de precisión y realizar la comparación de los mismos, de las técnicas en estudio se determinó que el SVM, es la técnica con mayor precisión para clasificar características de esta índole, mostrando un resultado del 96.875%, en segundo lugar tenemos a la técnica Fuzzy c-Means con un 95.455%, el tercer lugar es para la red neuronal probabilística con 93.75% y K-means con 87.879%, los investigadores resaltan que en el futuro se pueden probar diferentes métodos de aprendizaje automático alineados a la clasificación para una mejor precisión en enfermedades de la depresión para un tratamiento oportuno y temprano, ayudando así a bajar los altos índices reportados a nivel mundial.

En el artículo titulado “*Sentiment analysis of social networking sites (SNS) data using machine learning approach for the measurement of depression*”, publicado por los autores (Hassan, Hussain, Hussain, Sadiq, & Lee , 2017), hablan sobre el tema de la depresión y de cómo encontrar el nivel que cada persona tiene, extrayendo y observando emociones descritas en textos. Para el presente estudio se ha utilizado teorías de emociones, técnicas de aprendizaje automático y técnicas naturales de procesamiento del lenguaje en diferentes plataformas de redes sociales.

El artículo estudia las técnicas de clasificación de sentimientos, técnicas de clasificación binaria y técnica de clasificación de sentimientos de clases múltiples, para el caso en la técnica de clasificación binaria, cada uno se clasificó en la categoría  $C$  donde  $C = \{\text{Positivo}, \text{Negativo}\}$  documento en  $D$  donde  $D = \{d_1, d_2, d_3, \dots, d_n\}$  son clases y en clasificación de sentimiento de clases múltiples el  $d_i$  se clasifica en la categoría:  $C = \{\text{StrongPositive}, \text{Positive}, \text{Neutral}, \text{Negative}, \text{StrongNegative}\}$ .

En nuestra metodología propuesta, hay cuatro componentes principales que son el preprocesamiento, la extracción de características, el meta aprendizaje y los datos de capacitación. Para realizar el procesamiento se divide el párrafo en oraciones y luego dividimos las oraciones en palabras, posterior a ello eliminamos las palabras inútiles que no transmiten mucho significado.

En el caso de la extracción de características está relacionada con una bolsa de opiniones en base a léxicos y objetivos utilizando diccionarios, frases, adjetivos sustantivos y adverbios.

Por otro lado, para el meta aprendizaje se combina varios clasificadores eligiendo las etiquetas que tenga más votos, La votación es un conocido procedimiento de agregación que combina diferentes opiniones de los clasificadores de votantes en consenso Aquí usamos tres clasificadores, Máquina vectorial de soporte (SVM), Bahías Ingenuas (NB) y Máxima Entropía (ME) dichos algoritmos realizan funciones de predicciones de acuerdo a las etiquetas obtenidas, el  $d_i$  se clasifica en la categoría:

$C=\{\text{StrongPositive, Positive, Neutral, Negative, StrongNegative}\}$ , hay cuatro componentes principales que son preprocesamiento, StrongNegative}.

En nuestra metodología propuesta, hay cuatro componentes principales que son el preprocesamiento, la extracción de características, el meta aprendizaje y los datos de capacitación. Para realizar el procesamiento se divide el párrafo en oraciones y luego dividimos las oraciones en palabras, posterior a ello eliminamos las palabras inútiles que no transmiten mucho significado.

En el caso de la extracción de características está relacionada con una bolsa de opiniones en base a léxicos y objetivos utilizando diccionarios, frases, adjetivos sustantivos y adverbios.

Por otro lado, para el meta aprendizaje se combina varios clasificadores eligiendo las etiquetas que tenga más votos, La votación es un conocido procedimiento de agregación que combina diferentes opiniones de los clasificadores de votantes en consenso Aquí usamos tres clasificadores, Máquina vectorial de soporte (SVM), Bahías Ingenuas (NB) y Máxima Entropía (ME) dichos algoritmos realizan funciones de predicciones de acuerdo a las etiquetas obtenidas.

En este documento hemos hecho una comparación entre clasificadores SVM, NB y ME con respecto al análisis del sentimiento del nivel de oración para la medición de la depresión así mismo Hemos adoptado el modelo de votación y la técnica de selección de características, de lo expuesto se menciona que Observamos que la precisión de SVM es del 91 %, la precisión de la base nave es del 83 % y la precisión de la entropía máxima es del 80%.

Los autores (Baek & Chung, 2020), publicaron un artículo titulado “*Context Deep Neural Network Model for Predicting Depression Risk Using Multiple Regression*” donde se aborda el tema de depresión y enfrenta el problema de predecir situaciones y entornos que influyen en la depresión en consideración de la información de contexto.

El presente estudio propone el modelo de redes neuronales profundas (DNN) para predecir el riesgo de depresión mediante regresión múltiple. El modelo propuesto predice el contexto de riesgo de depresión en combinación con la información de contexto y de DNN.

El modelo de redes neuronales profundas (DNN), consiste en la información para predecir situaciones y entornos que influyen en la depresión en consideración de la información de contexto. Cada información de contexto relacionada con las variables predictoras de depresión se convierte en una entrada de DNN, y la variable para la predicción de la depresión se convierte en una salida de DNN. Para la conexión DNN, se utiliza el análisis de regresión para predecir el riesgo de depresión con el fin de predecir el contexto potencial que influye en el riesgo de depresión.

Según la evaluación del rendimiento, el modelo propuesto tenía el mejor rendimiento en el análisis de regresión. Para el desarrollo, como primer paso fue la organización de datos y el procesamiento previo. Para la organización de los mismos, se utiliza una encuesta realizada dentro del campo de la salud y nutrición. Según el modelo se inicia con la selección de Variables Utilizando análisis de regresión múltiple, luego realizamos el procesamiento de datos que se utilizara para predecir el riesgo de la depresión de un usuario de acuerdo con enfermedades físicas o condiciones externas y por último realizaremos la predicción de contexto utilizando el modelo de la red Neuronal profunda.

El DNN de contexto propuesto llevó a cabo cuatro evaluaciones de rendimiento. En primer lugar, como resultado de la evaluación de la conformidad, el análisis de regresión múltiple evaluó la potencia explicativa de la variable como no apta, y el DNN se evaluó más inadecuado debido a la menor función de pérdida y precisión que el Context-DNN. En segundo lugar, se llevó a cabo la validación cruzada para evaluar el modelo de Context-DNN. En tercer lugar, la precisión y la retirada se compararon con el DNN general.

Como resultado, el rendimiento de Context-DNN obtuvo un 90% de precisión el mismo que fue mejor que el de DNN general que solo obtuvo un 80%. El modelo propuesto aprendió DNNs individualmente para mejorar la precisión y utilizó variables predictoras a través del análisis de regresión. Por lo tanto, el modelo de DNN de contexto propuesto permite juzgar el contexto de uno para el riesgo de depresión con precisión, para seguir cuidando la salud mental y para prevenir la depresión.

En el artículo *“Depression Detection using Emotion Artificial Intelligence”* publicados por los autores (Deshpande & Rao, 2018), enfrentan el tema de depresión, donde resumen que dicha enfermedad aumenta el riesgo de muerte prematura. Además, es una de las principales causas de ideación suicida y provoca un importante deterioro de la vida cotidiana de las personas. Ante el problema expuesto los autores están desarrollando un método donde se usa la tecnología de inteligencia artificial con el uso de los algoritmos de máquinas de vectores de apoyo y clasificadores Naive-Bayes. Para el desarrollo del presente método se inicia con el proceso de recolección de datos con ayuda de una Interfaz de programación de aplicaciones (API) de la red social Twitter, para generar los datos a necesitar, se realiza el procesamiento del lenguaje natural el cual facilita una clasificación mucho mejor que el análisis sentimental realizado por el ser humano. Luego de realizar el procesamiento para la recolección de los datos se procede a realizar la eliminación de palabras de paso y palabras de parada, que son aquellos textos que no tienen relevancia para el estudio. De los textos rescatados validos se selecciona las partes importantes del texto que se utilizará.

A continuación, el clasificador de texto se entrena con los datos de texto procesados de Twitter, en esta fase de prueba, la predicción de la clase se realiza en conjunto de datos donde se identifica posibles Tweets que demuestren tendencias de depresión.

Después de identificar los casos de depresión, los algoritmos de clasificación requieren de dos parámetros, la agrupación de entrenamiento

y la etiqueta. Para este caso, el conjunto de entrenamiento es el conjunto de tuits que deben ser procesados para alimentar un clasificador, así mismo debe convertirse en un formato vectorial para su posterior procesamiento. El conjunto de etiquetas correspondientes a cada tuit también se introduce en el clasificador en forma de vector Guardando el clasificador y el objeto vectorizado de conteo: Dado que el entrenamiento debe hacerse una vez, el objeto clasificador entrenado debe ser cargado en un archivo pickle. Lo mismo ocurre con el objeto Count Vectorizer. Por lo tanto, ambos objetos son volcados en un archivo pickle para su uso posterior.

Para obtener los resultados se evalúan en base a la función de la puntuación F1 y la precisión. La puntuación F1 es la principal medida de rendimiento y la precisión es la medida secundaria. La puntuación F1 se calcula a partir de la precisión y la recuperación En este caso, P significa Precisión y R es la Recuperación se detalla en la siguiente fórmula matemática.

$$F1\ score = 2 * \frac{P1 * R1}{P1 + R1}$$

En este caso, P1 significa Precisión y R1 es la Recuperación.

De los resultados se desprende que Multinomial Naive Bayes ha sido el que mejor ha funcionado, con una puntuación F1 de 83,29%, mientras que SVM ha obtenido una puntuación F1 inferior, de 79,73%. La Precisión y la Recuperación siguen la misma tendencia y el Bayes ingenuo multinomial supera al SVM. La matriz de confusión reporto como varios parámetros como verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos. Su precisión del Mutinomial Naive Bayes es del 83% y del 79% en el caso de la SVM.

Los autores (Allen , y otros, 2017, págs. 1-4) en su investigación “Machine Learning Aided Prediction of Family”, abordó el tema de la depresión, enfocados a Predecir si un individuo tiene una historia de depresión familiar.

Los autores proponen un enfoque de aprendizaje automático basado en datos que utiliza características cerebrales multimodales, En el desarrollo

de este estudio se ha utilizado características cerebrales multimodales, como por ejemplo morfometría de la materia gris basada en imágenes ponderadas y conectoma estructural basado en tractografía de difusión probabilística, esto con la finalidad de capturar las representaciones complejas de la estructura cerebral, bajo la información recolectada se implementaron la regresión logística (LR) con regularización, la máquina de vectores de apoyo (SVR) y la red neural convolucional de gráficos (GCN). El método está compuesto por dos procesos, una que trata de recolectar el nivel de riesgo de cada participante se definió en función de la primera generación (G1). La descendencia de la G1 (generaciones 2 y 3, G2 y G3) recibió una etiqueta de alto riesgo si el progenitor de la G1 tenía antecedentes de trastorno depresivo mayor (TDM). Para el presente estudio, se recogieron y obtuvieron las resonancias magnéticas de 93 participantes de las muestras G2 y G3, y cada participante fue etiquetado como 0-bajo o 1-alto riesgo.

Para extraer las características derivadas de las imágenes cerebrales estos se analizaron mediante Imagen por Resonancia Magnética (IRM), a través de las imágenes estructuradas se ha obtenido información morfométrica regional del cerebro, como el grosor, el área y el volumen, realizando una segmentación y parcelación automatizada del cerebro utilizando la suite de análisis de imágenes Freesurfer v6.0, a partir de las imágenes de difusión, obtuvimos matrices de conectividad de la materia blanca utilizando tractografía probabilística y un algoritmo para filtrar las estimaciones de tractos falsos positivos. Estas matrices de conectividad se ponderaron utilizando dos métodos diferentes: el número de estimaciones de tractos y su longitud media. Las características del cerebro se normalizaron primero para mantener los datos en un rango comparable, y segundo para mantener las características ponderadas por igual. Las características normalizadas también suelen mejorar la eficiencia de los algoritmos de aprendizaje automático.

Cada modelo utilizó una validación cruzada estratificada de 10 veces. En el caso de GCN, se utilizó un conjunto de validación para ajustar los hiperparámetros. Sin embargo, tanto la regresión logística como la SVM

pueden ajustarse para el parámetro de regularización en un conjunto de validación.

Dado que la regresión logística y la SVM no son tan propensas al sobreajuste como las redes neuronales, sólo se necesitó una agrupación de prueba y otro de entrenamiento, aunque los conjuntos de datos se crearon mediante validación cruzada KFold. Cada conjunto de datos se rotó cada vez que se entrenaron los algoritmos. En su gran mayoría los modelos se repitieron diez veces con una validación cruzada estratificada para tener en cuenta las fluctuaciones en la precisión de acuerdo a la inicialización aleatoria de los pesos en cada modelo. Se calculó la precisión media y la desviación estándar de las 10 ejecuciones para cada modelo.

Se trazó una curva final de características operativas del receptor (ROC) para cada uno de los modelos luego se realizó el cálculo del área bajo la curva, el mismo que evalúa la fuerza de predicción. Las curvas ROC muestran su vínculo entre la especificidad y la sensibilidad de un modelo. Además, dado que utilizamos una regresión logística, los pesos de las características se extrajeron fácilmente y luego se interpretaron. Por ejemplo, el peso más alto o la distribución del histograma de pesos influyen en la precisión de la predicción.

Del estudio se obtuvieron resultados donde cada participante tenía más de 15.000 rasgos cerebrales, de los que se eliminaron todos los rasgos nulos y duplicados mediante un preprocesamiento de datos que redujo el número total de rasgos a 7876 rasgos utilizables, posteriormente las características se clasificaron en función de la correlación entre cada característica y el riesgo de depresión. Para normalizar los datos, se tomó una puntuación z para garantizar que la entrada en los modelos de aprendizaje automático fuera eficiente y precisa. Las curvas ROC están diseñadas para representar el compromiso entre la sensibilidad y la especificidad, por lo que se construyeron curvas para cada modelo con el fin de visualizar el poder de predicción.

En este estudio, los resultados fueron impresionantes, ya que la regresión logística tuvo un área bajo la curva (AUC) del 99%, lo que deja muy poco

margen de mejora. La SVM, con un AUC del 98%, y la GCN, con un AUC de El 96% también obtuvo resultados prometedores. La elevada precisión de la clasificación (validada de forma cruzada) (96,78%) muestra el significativo impacto de la depresión familiar en la estructura cerebral y el conectoma. El estudio expuesto servirá de base para el desarrollo de un modelo clínicamente útil para la predicción del riesgo de psicopatología utilizando fenotipos derivados de imágenes cerebrales.

Según, (Choudhury, y otros, 2019, págs. 789-794), en su artículo *“Predicting Depression in Bangladeshi Undergraduates using Machine Learning”*, explican el problema de la depresión y su impacto en los estudiantes de la universidad de Bangladesh, los jóvenes con esta enfermedad resultan perturbados en su salud emocional, con la presencia de diversos síntomas como ansiedad, tristeza, pensamientos suicidas, aumento o pérdida de peso entre otras, que persisten durante el periodo mínimo de dos semanas, razón por la cual los escritores buscar predecir la depresión y derivarlo a algún psiquiatra para su atención psicológica y clínica. Para el estudio se aplicaron diferentes encuestas para la recolección de datos que consta dos secciones, para la primera, se especifica datos básicos sobre el participante, y en la segunda sección se plasma interrogantes sobre las características más relevantes de la depresión, la respuesta tiene una duración de 15 a 20 minutos, donde contestarían preguntas basados en los sentimientos actuales dentro del rango 7 a 15 días, con las recomendaciones de hablar con verdad y marcar sin pensarlo mucho. De la encuesta se logró recolectar 935 respuestas el cual permitió dar inicio a la construcción del modelo de aprendizaje automático. Para ello se inició el proceso con la limpieza de datos quedando 800 registros, con la condición si el estudiante tiene o no depresión, guardándose en un archivo con extensión CSV, luego de ello basados en el resultado se aplicó tres algoritmos con la finalidad de evaluar la presión y otras métricas de interés. Los algoritmos utilizados en el modelo fueron los Kvecinos más cercanos (KNN), Random Forest (RF) y maquinas vectores de soporte (SVM), que a partir de una matriz de confusión se ha realizado los cálculos de precisión teniendo como resultado que, KNN 67%, SVM 73% y RF 75 %,

comprobando que el algoritmo RF es el más eficiente para la predecir la depresión, por otro lado, los escritores informan que no ha realizado muchos trabajos y necesita una atención por los investigadores.

### 1.3. Teorías relacionadas al tema.

Según (González & Valls, 2015), afirman que en toda revisión de la literatura debe empezar desde el inicio de la investigación, toda vez que ayudará a describir el desarrollo de una perspectiva teórica de las diferentes actividades relacionadas al tema.

#### 1.3.1. Depresión y su guía clínica

Este diagnóstico en un estado crítico puede llegar a culminar con el suicidio. (Vargas Cajahuanca, Gallegos Cazorla, Salazar de la Cruz, Bonilla Untiveros, & Caballero Ñopo, 2019)

Según (Bellido Riquelme , Aragonés Alosilla, Bellido Riquelme, Castro Paredes, & Ramírez Freyre, 2008, pág. 11) indican que la depresión se encuentra incluidos en los trastornos afectivos, y se plasman en la guía de clasificación internacional de enfermedades, décima edición (CIE - 10), en dicho documento los episodios enfocados a la depresión es el código F32, los mismos que son clasificados de la siguiente manera.

Tabla 1.

*Codificación de la depresión según CIE - 10*

Código	Descripción
F.32.0.	Suceso depresivo leve
	F.3 2.00. sin síntomas somáticos
	F.3 2.01. con síntomas somáticos

*Nota.* Codificación sobre depresión Fuente: Guía CIE – 10.

#### 1.3.2. Signos y síntomas

Según (Bellido Riquelme , Aragonés Alosilla, Bellido Riquelme, Castro Paredes, & Ramírez Freyre, 2008, pág. 13), indican los síntomas signos que están relacionado con la depresión.

**a) Tristeza o Nostalgia:** ánimo superior que obstaculiza de forma negativa la vida cotidiana, de manera social, familiar y sexual, tanto que causa la idea de no poder vivir y que es incapaz de dar amor y afecto lo que conlleva a subir su sensibilidad culpa.

**b) Desgano:** el paciente se vuelve desidioso, sin voluntad de seguir viviendo.

**c) Ansiedad:** es la escolta cercana del deprimido, si este predomina se concluye que el paciente se encuentra en una depresión ansiosa, de lo contrario estaríamos frente a una depresión inhibida, estos en lo particular lo tienen las personas agresivas, resentidas, coléricos.

**d) Insomnio:** Dificultad para dormir, levantándose muy temprano, con una actitud de mal temple.

**e) Alteraciones del pensamiento:** alucina que tiene muchas enfermedades, así mismo imagina sentimientos de derrota con fuertes ideas de culpa y obsesiones.

**f) Alteraciones somáticas:** Generalmente se cuenta con una insistente emoción de agotamiento, por lo que ocurren dolores crónicos como sudoración nocturna y estreñimiento.

**g) Alteraciones en el comportamiento:** cambios bruscos en el comportamiento, puede estar tranquilo(a) o estallar en llanto sin algún motivo, violento ataque de crisis, entre otros. Al paciente le cuesta tomar decisiones y cae en el rendimiento en sus funciones cotidianas.

**h) Alteraciones de apetito y de peso:** en su gran mayoría los pacientes pierden peso por la ausencia del apetito.

**i) Perdida del placer:** a nivel sexual en el caso de la mujer frigidez y en el varón impotencia, también desaparece o disminuye la alegría en lo laboral, actividades deportivas y otras que resultaban satisfactorios.

**j) Culpa excesiva:** el paciente carga con una culpa que no existe lo cual puede terminar en delirios.

**k) Pensamiento suicida:** se presenta altos índices de preocupación por la muerte y acogen pensamiento de autodestrucción.

**l) Disminución de energía:** los pacientes se tienen más cansados por las mañanas que por las tardes esto de manera injustificada.

### 1.3.3. Diagnóstico de depresión

Las reglas para este punto están descritas de acuerdo al CIE – 10, dándose mayor énfasis en los síntomas de depresión, confirmado por más de dos semanas o periodos cortos en caso los síntomas sean graves o bruscos desde un comienzo. Para el reconocimiento y diagnóstico en los adolescentes y niños, los síntomas descritos en el punto anterior siempre a menudo tienden a confundirse con otros problemas físicos y de conducta, por lo cual se ha creado un cuadro comparativo describiendo lo mencionado (Dirección Regional de Salud Cusco, 2018).

Tabla 2.

*Diagnóstico de la depresión en jóvenes y niños.*

<b>Síntomas depresión</b>	<b>Observación en los jóvenes y/o niños</b>
Todo el día se encuentra en un estado de ánimo depresivo	Ánimo malhumorado e irritable
Disminuye el interés de actividades favoritas.	Desinterés en video juegos, deportes y actividades que realiza de manera frecuente con amigos.
Perdida o incremento de peso	Frustración por el aumento de peso a consecuencia de bulimia o anorexia, dolor de estómago, dolor de cabeza.
Hipersomnio o insomnio	Vicio por la televisión hasta la madrugada y negarse a levantarse para cumplir con sus actividades.

Retraso psicomotor o agitación	Mencionar que se ira de casa o tener la valentía de hacerlo.
Perdida de energía / fatiga	Fastidio, cansancio de manera permanente.
Culpa / baja autoestima	Conducta negativa
Disminuye la concentración, indecisión	Falta constantemente y no tiene un rendimiento académico optimo.
Comportamiento / ideas suicidas	Conducta e ideas de suicidio de manera frecuente

*Nota:* Esta tabla muestra los principales síntomas de depresión y el modo de observar a los niños y adolescentes. Fuente: (Dirección Regional de Salud Cusco, 2018).

#### 1.3.4. Factores de riesgos asociados

según (Bellido Riquelme , Aragonés Alosilla, Bellido Riquelme, Castro Paredes, & Ramírez Freyre, 2008), afirma que, existe dos factores que a continuación se detalla:

a) **Estilos de vida:** la falta de efectividad y estabilidad puede liberar la presencia de la depresión, esto en la etapa de infancia o adolescencia. Si existe una familia comunicativa y coherente disminuye los factores del riesgo expuesto, mientras que, en una familia disconforme, la probabilidad es más alta a la depresión. Los casos más comunes se dan por la inestabilidad emocional principalmente de los padres, ya que se encuentran en discusiones y continuos problemas. El soporte de entorno familiar debe ser ameno y positivo para poder mejorar este trastorno mental que afecta a cada integrante de la familia.

b) **Factores hereditarios:** Se cuenta con un alto índice de probabilidad de tener depresión, si existe un historial clínico de la enfermedad en la familia, lo que significa que puede haber una herencia biológica. El riesgo puede ser mayor si los pacientes cuentan con un trastorno bipolar, sin embargo,

no todas las familias lo poseen. A todo lo expuesto se menciona que existe factores adicionales que causan la depresión, estos pueden ser factores ambientales o bioquímicos que desarrollan estrés y otras circunstancias psicosociales.

### **1.3.5. Auto escala de Zung para depresión**

Ante la llegada de diversas escalas de clasificación psiquiátrica en la identificación de la depresión, se plantean varias dudas acerca del por qué los pacientes con depresión perciben disforia emocional, teniendo síntomas inexplicables desde el conocimiento médico, los pacientes con este diagnóstico son evaluados inadecuadamente, ya que estos pacientes se caracterizan por presentar alexitimia, es decir con déficit emocional comunicativo que se asocia al pensamiento operacional. (Piñerúa Shuhaibar, Estévez, & Suárez Roca, 2020)

El diagnóstico del trastorno mental psiquiátrico, reportado de manera frecuente después de una consulta médica, es la depresión, algunos autores han considerado la somatización como característica única del cuadro depresivo y su inclusión como criterio de identificación de la misma (Piñerúa Shuhaibar, Estévez, & Suárez Roca, 2020).

La auto escala de Zung, es considerado un instrumento para realizar la medición de la depresión a través de información de aspectos cognitivos, afectivos y somáticos del paciente, dicha documento tiene una especificidad del 63% y sensibilidad del 97%, aprobando un acierto del 82% para la discriminación de la depresión, así mismo están relacionadas a los cambios autonómicos y sensoriales de los pacientes deprimidos. Para estos casos se han creado preguntas y respuestas sensoriales para realizar el descarte si un paciente tiene o no depresión (Piñerúa Shuhaibar, Estévez, & Suárez Roca, 2020).

Tabla 3.

*Escala de autovaloración según Zung, para discriminar la depresión*

<b>Categoría o síntoma</b>	<b>Pregunta</b>	<b>Numero pregunta</b>
<b>I. Afectiva</b>		
1. Tristeza, depresión	¿me siento desalentado, triste, y desmotivado?	(1)
2. Desmotivación y melancolía.		
3. Llanto	¿tengo llantos continuos o razones para hacerlo?	(3)
<b>II. Fisiológica</b>		
1. Variaciones diarias: síntomas en exceso por la mañana y alivio durante el día.	¿en horas de la mañana me siento bien?	(2)
2. Siesta: levantarse temprano muy frecuente.	¿puedes dormir durante la noche?	(4)
3. Hambre: reducción en el consumo de alimentos	¿cómo más que antes?	(5)
4. Decrecimiento de peso: Falta de apetito o falta de sueño.	¿noto que perdí peso?	(7)
5. Sexualidad: libido reducida.	¿gozo cuando, hablo o estoy con varones/mujeres cautivadoras?	(6)
6. Gastrointestinal: estreñimiento	¿me estreñimiento frecuentemente?	(8)

---

7. Cardiología: taquicardia	¿mi corazón tiene latidos acelerados?	(9)
8. Osteomuscular: cansancio inexplicable.	¿siento cansancios inoportunos?	(10)
<b>III. Psicomotora</b>		
1. falta de respiración	¿tengo intranquilidad sin control?	(13)
2. Dilacion Mental	¿me resulta fácil hacer, las cosas que hice algún momento?	(12)
<b>IV. Psicológica</b>		
1. Desorden mental	¿tengo mi mente clara?	(11)
2. desacompañamiento	¿considero tener una vida agusta?	(18)
3. Desesperación	¿tengo un futuro con esperanzas?	(14)
4. Inseguridad	¿me resulta sencillo tomar mis decisiones?	(16)
5. Disforia	¿tengo mal humor siempre?	(15)
6. Insatisfacción	¿deleito las cosas que solía hacer?	(20)
7. Desvalorización personal	¿percibo que soy útil y que me necesitan?	(17)

---



*Nota:* Valores de respuestas que serán comparados para definir los el grado de depresión.

#### **4.1.1. Lenguajes de programación (LP)**

Las razones principales para que las personas aprendan leguajes y/o técnicas de programación, es manejar el computador como una herramienta para resolver problemas. Es necesario definir los siguientes pasos: análisis del problema a resolver, diseñar un algoritmo, transformar el algoritmo a un programa informático y la ejecución y validación del mismo, conllevando así a desarrollar automatizaciones de procesos informáticos para el procesamiento de información (Joyanes Aguilar, 1996).

##### **a) Lenguaje de Maquina**

Son aquellos que se encuentran redactados en lenguajes intangibles por las maquinas (computadoras), ya que las instrucciones que poseen se encuentran en cadenas binarios, es decir en 0 y 1, así mismo las posiciones (dirección) de memoria se les llama código de máquina u instrucciones de máquina. Todo lo relacionado a lo expuesto dependen del hardware de la computadora, por lo que existirá diferencia entre una y otra. (Joyanes Aguilar, 1996).

##### **b) Compilación y sus fases**

Es el procedimiento que traduce el programa fuente (PF) a programa objeto (PO), el PO obtenido de esta compilación ha sido traducido a un lenguaje de máquina, y por ende se consigue el programa de maquina real a través de ejecutable. (Joyanes Aguilar, 1996).

##### **c) C / C++**

El lenguaje de programación (LP) C, es uno de LP de mayor propósito, dentro de sus características se indica que es simple, popular y flexible, además es un LP estructurado de manera independiente que se utiliza para el desarrollo de diferentes aplicaciones, bajo diferentes plataformas (Windows, Linux y

otros), el mencionado LP tiene la arquitectura hasta para escribir programas de mayor complejidad como Git, base de datos, el intérprete Python y mucho más. (Pacheco Patiño, 2019).

El LP C++, es el más popular ocupando el cuarto lugar según la clasificación del año 2016 considerado como un LP principal del IEEE. Conocer este LP es una de las elecciones más inteligentes que todo profesional debería tener en cuenta ya que con ello se desarrolla de manera moderno juegos, navegadores, sistemas operativos, sitios web, productos asistidos por computador y de diseño gráfico (Pacheco Patiño, 2019).

#### **d) C#**

El lenguaje de programación (LP) C#, en la actualidad es uno de los más populares del internet, se considera uno de los mayores lenguajes de alto nivel que pertenecen al compendio .NET, el mencionado LP es una versión avanzada de los LP C y C++, con este se puede realizar proyectos convencionales para el internet. Dentro del compendio .NET se incorpora un compilador que hace común para todos los LP (Ceballos Sierra, 2008).

#### **e) Java**

El lenguaje de programación (LP) Java, hoy en día es utilizado en diversos ambientes, insertándose programas en páginas web, los cuales son llamados applets, a consecuencia de ello Java tuvo su popularidad y llegó a posicionarse en uno de los lenguajes de programación líderes en todo el mundo. Los applets aún tienen un rol importante en cuanto a la popularidad de este LP, sin embargo, la implementación de la aplicación de servlet, aplicación Java Server Page (JSP), aplicación Java Micro Edición (ME) y Java Edición Estándar (SE), hacen que el LP sea más aceptado por bastantes usuarios de internet (John & Raymond, 2009).

En cuanto a las características que este LP, este cuenta con un lenguaje compilado e interpretado, generando código Bytecodes, los mismos que son interpretados por una máquina virtual, este LP hoy en día es un lenguaje

orientado a objetos que es utilizado para desarrollar diferentes tipos de proyectos, en cuanto a su sintaxis se asemeja a los LP C y C++. (Belmonte Fernández, 2020).

**f) Python**

Excelente Lenguaje de programación (LP), novedosa y favorita por la mayoría de los programadores a nivel del Internet, promete facilidad para su aprendizaje y contiene estructuras eficientes de datos de más alto nivel, además dicho entorno de desarrollo está orientada a objetos lo que hace la escritura y sintaxis más elegante, desarrollándose bajo cualquier sistema operativo, el intérprete de este LP y su biblioteca se encuentran disponible de forma gratuita y en formato binario para todas sus plataformas . (Biblioteca Estandar de Python, 2021)

**g) R**

Se menciona que R es un lenguaje de programación de distribución libre con licencia GNU (Sistema operativo Unix), vinculados al campo de la gráfica y estadística, el mencionado lenguaje se ejecuta en todas las plataformas con el propósito de emplear diversas técnicas y/o algoritmos estadísticos. Dentro de sus principales características se encuentra que cuenta con paquetes modulares para su ejecución y el sistema base de R, el mismo que incluye Utilidades, base de datos, herramientas entre otros (Santana Julio & Mateos Farfan, 2014).

Otra de las definiciones de este lenguaje de programación, es que **R**, tiene integrado un conjunto de funciones para la manipulación de información estadística y su visualización gráfica, también está orientado a objetos lo que hace que este sea potente, flexible y posea un corto tiempo para su aprendizaje, en la actualidad se cuenta con más de 15,303 paquetes implementado que cubren la materia de los campos a estudiar (Jimenez J. U., 2019)

#### **4.1.2. Algoritmos de predicción**

Son aquellos algoritmos estructurados que permite evaluar diferentes probabilidades a partir de la declaración de una serie de valores, el mismo que según su entrenamiento podrá determinar un aprendizaje computacional supervisado o no supervisado para definir suposiciones positivas, entre las posibles predicciones a futuro (Timón, 2017).

#### **4.1.3. Inteligencia Artificial (IA)**

Se entiende como tal, a la disciplina que se encuentra ligada a la computación cuyo objetivo es simular ciertas habilidades de inteligencia humana en sistemas inteligentes. Se entiende con inteligencia humana, a las diversas percepciones como la visión, audición, entre otros, se debe exponer que un sistema inteligente es aquel que utiliza herramientas de cálculo estadístico, numérico para el proceso de actividades automáticas, como por ejemplo la robótica. Es por ello que la IA contribuye notablemente a los estudios científicos, matemáticos y técnicos. En estos nuevos tiempos la IA permite desarrollar aplicaciones en los dispositivos electrónicos, aportando notablemente en el mundo de la microelectrónica. (Benitez, Escudero, Kanaan, & Masip, 2018).

#### **4.1.4. Minería de datos**

Es el termino genérico este representa a grandes volúmenes de datos, ya sea de tipo texto, imágenes, audio, video, etc. Estos alimentados por diferentes fuentes de la internet, dispositivos electrónicos, sistemas de información de instituciones del sector público o privado. En la actualidad esta nueva tecnología ha tenido una acogida impresionante puesto que lo datos se ha convertido en materia prima que se pueden explotar según el contexto de estudio, con ellos se puede explotar y tener conocimientos de estimaciones a futuro, gustos, etc., los mismos que ayuda a elegir las mejores decisiones en cualquier empresa (Ferri & Ramírez, 2008).

#### **4.1.5. Aprendizaje automático**

También conocido como Machine Learning (LM), cuyo fin es crear aplicaciones capaces de procesar cualquier tipo de información a partir de

datos que no se encuentran estructurados incluidas como ejemplos. (Herrera, A. & Duran, J., 2016).

#### **4.1.6. Tipos de aprendizaje automático**

##### **Aprendizaje no Supervisado**

Caracterizado por encontrar patrones o atributos que permitan categorizar y separar datos de diferentes conjuntos de manera que este nos pueda decir si una imagen, corresponde a un animal o de un gato, entre otras características. (Palma & Marín, 2008)

##### **Aprendizaje Supervisado**

Son conocidos por contar con algoritmos que tienen un conjunto de ejemplos, que clasifica de acuerdo a las variables de ingreso, el algoritmo para este caso se entrena con información histórica el mismo que ayuda a su aprendizaje para su estimación en las variables de salida. (Mesenguer & Lopez, 2017).

#### **4.1.7. Predicción**

Se define como tal, al aprendizaje entrenado para predecir el futuro, utilizando patrones de comportamiento para la aplicación a cualquier evento conocido que se requiera estudiar (Timón, 2017).

#### **4.1.8. Algoritmos de clasificación**

##### **K -Vecinos Más Cercanos (K - NN)**

Algoritmo de aprendizaje supervisado que se enfoca en la clasificación, La lógica demuestra que es necesario almacenar un conjunto de entrenamiento, el mismo que para clasificar una nueva instancia, esta procura encontrar casos similares asignándole a la clase más probable de estos. El presente método en estudio estuvo representado mediante una ecuación de la siguiente manera. (Ihsan A, 2017)

$$d \left( \begin{matrix} i \\ j \end{matrix} ; \begin{matrix} i \\ q \end{matrix} \right) = \sqrt{\sum_{k=1}^{|A|} (a_{kj} - a_{kq})^2}$$

En donde:

k = prototipo cerca a la instancia.

iq = distancia euclidiana

ij e iq = distancia entre las instancias

Entrenamiento:

Cada caso Tr, se agrega a una lista de prototipos de entrenamiento.

Clasificación:

Cada prueba a ser clasificada, qi,

Sea  $i_1, \dots, i_k$  los k prototipos de la lista, son los cercanos a iq.

Retorna.

$$ca = \arg \max_{i \in C} \sum_{j=1}^k \delta(c_i, c_j)$$

Donde:  $\delta(a, b) = 1$  si  $a = b$  y  $\delta(a, b) = 0$ , en caso diferente.

El algoritmo estudiado no genera un modelo fruto de los datos de aprendizaje en el momento que empiece a clasificar una instancia nueva, es por ello que es llamado método perezoso de aprendizaje, en su funcionamiento del algoritmo se realiza un cálculo de la distancia con todas instancias que se ha entrenado y de estas se seleccionas las instancias más cercanas. La nueva instancia lo determinamos como una clase mayoritaria entre sus K instancias más próximas. Para su aplicación se debe saber el siguiente proceso.

En primer lugar, fijamos los valores para K. recomendado pequeño, ahora dada una instancia y, del entrenamiento de datos, el algoritmo selecciona las k instancias de datos entrenados y entre ella elige la más cercana teniendo en cuenta su similitud métrica, posterior a ellos se adjudica la instancia y a la más frecuente entre los k más cercanos (Minguillón, Casas, & Minguillón, 2017).

## Naive Bayes

Una de las técnicas de clasificación estadística más amplias basadas en aprendizaje automático o de Machine Learning, todo fundamentado en el teorema de Bayes.

El modelo tiene funciones predictoras que son independientes entre sí, es decir no tienen relación directa dentro de un conjunto de datos una cierta característica. Además, se puede implementar modelos de una manera fácil debido a su simplicidad. (Lucio, Fernandez, & Martines, 2013).

Es indispensable mencionar que la técnica estudiada funciona mejor cuando los datos tienen una dimensionalidad alta, el mismo que es capaz de dar un resultado de salida más posible en función a la de entrada. (Manna, 2018)

Este método proporciona la siguiente ecuación:

$P1(c | x)$  en inicio de  $P1(c), P1(x), P1(x | c)$ :

$$q(c | x) = \frac{q(x|c)q(c)}{P1(c)}$$

Entonces:

$q(c | x)$ : La posibilidad de clase posterior (c, objeto) dado el predictor (x, atributos).

$P1(c)$ : La posibilidad previa de clase.

$q(x| c)$ : Posibilidad que es probabilidad del predictor de clase.

$q(x)$ : La posibilidad previa del predictor.

## Árboles de clasificación y regresión

Algoritmo estadístico que tiene alta precisión para predecir el valor de una variable categórica o continua en un conjunto de datos.

El método estudiado es una técnica de aprendizaje no paramétrica que crea árboles de clasificación o regresión, esto de acuerdo a si la variable es numérica o categórica.

Están formados por reglas en variables y modelado de datos, para que se obtenga una mejor división y poder diferenciar las observaciones enfocadas en la variable dependiente.

Una vez seleccionado se hace una división con una regla dividiendo un nodo en dos eso es efecto multiplicador por cada sub nodo, en procedimiento recursivo.

La división se detiene cuando el algoritmo detecta que ya no se pueden subdividirse o que se cumplan algunas reglas declaradas

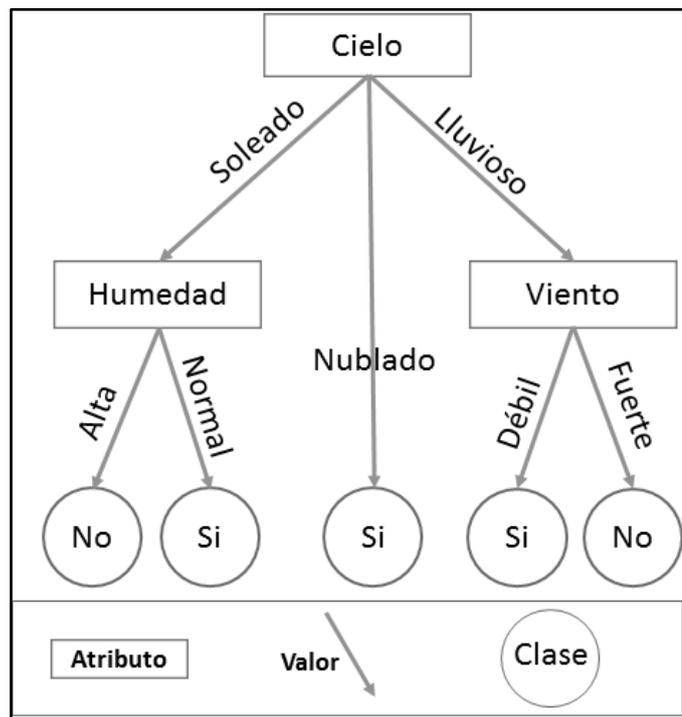


Figura 3. Árbol de decisión. Fuente:(Guallart Romeu, 2010).

Los principales procesos de división de nodos en sus atributos nominales se hacen con base a decisiones binarias, como por ejemplo usted es varón la contestación puede ser Si o No. Es por ello si uno de sus atributos tiene  $n$  valores posibles, entonces se generarán  $n$  valores particiones binarias entre los valores de  $1 \leq j \leq n$  (Benitez, Escudero, Kanaan, & Masip, 2018).

Para las particiones de atributos numéricos su rango es dividido en dos distancias  $z_i \leq b_c, z_i > b_c$  siendo  $b_c$  (punto medio de corte) una característica  $z_i$  en la evidencia  $E$ .  $b_c$  puede fijarse los índices adecuados para reducir el tamaño del árbol (Benitez, Escudero, Kanaan, & Masip, 2018).

## Redes neuronales artificiales (RNA)

Este algoritmo representa un modelo no lineal, que es capaz de formar funciones muy complejas, dicho método está dentro del universo de la IA, el mismo que según la arquitectura neuronal que este posea será la aplicación, el presente algoritmo puede ser usado para el reconocimiento de diversos patrones, clasificación, reducción en las dimensiones y las comprensiones de todo tipo de información.

Las redes neuronales son utilizadas cuando se desconoce el origen exacto entre aquellos valores de ingreso y de salida. Uno de los atributos importantes en las redes neuronales es que asimilan la relación a través de los atributos de entrada y salida basados en un ejercicio. Se conoce que hay 3 tipos de adiestramiento en redes neuronales que son, el aprendizaje por refuerzo, supervisado y no supervisado, siendo el aprendizaje supervisado el más común. (Timón, 2017).

Para poder entender mucho mejor el funcionamiento del algoritmo de la red neuronal es importante saber que, cada procesamiento de datos biológicos es vigoroso y flexible a los fallos. Además, es capaz de hacer su trabajo con datos incompletos, inconsistente o con ruidos, es por ello que la maquinaria que ejerce estas funciones es totalmente pequeña, compacta, paralela y difumina muy poca energía. (Ferri & Ramírez, 2008).

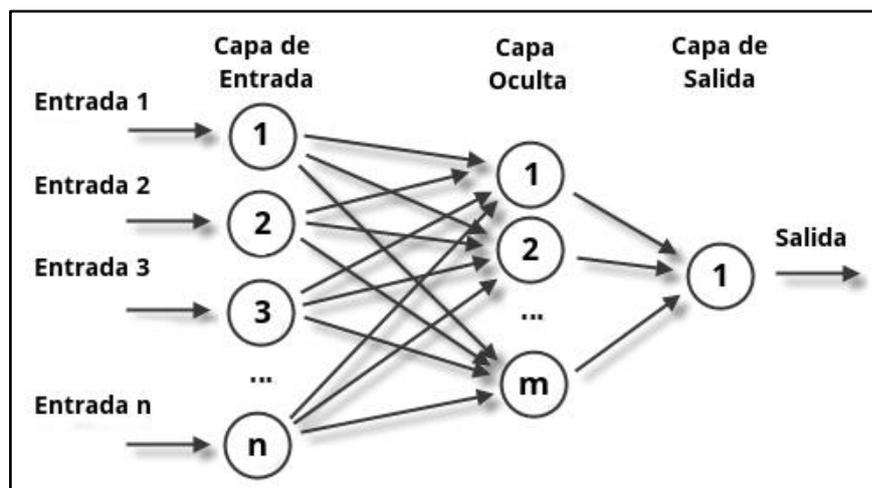


Figura 4. Modelo de funcionamiento estructural. Fuente: (IBM, 2020)

Cabe señalar que en la figura 4, se detalla los tres tipos de capa, la capa de entrada que es la encargada de recolectar datos de fuentes externas, la capa

oculta que quienes son los encargados de compartir el mismo tipo de información sin tener una conexión con el exterior y la capa de salida que transporta el resultado requerido hacia el exterior. (Caicedo & López, 2017). Para modelar redes neuronales definimos:

$$h(a + 1) = h(a) + \Delta h(a)$$

Donde,

$h(a + 1)$ : dato peso sináptico.

$h(a)$ : dato vigente del peso sináptico.

$\Delta h(a)$ : Variación del peso sináptico.

Una RNA no funciona con ciertos desvíos locales, se realiza un desvío global, que es calculado en base al error cuadrático promedio (Caicedo & López, 2017).

$$K_p = \frac{1}{2P} \sum_{p=1}^P \sum_{j=1}^k (d_{pj} - y_{pj})^2$$

Donde,

$K$ : Totalidad de Neuronas en el proceso, capa de salida.

$P$ : Totalidad de modelos en entrenamiento.

### **Support Vector Machine (SVM)**

El algoritmo estudiado, han demostrado un excelente rendimiento en los problemas de clasificación y predicción, dicho algoritmo minimiza el error de la generalización y los errores en nuevas instancias.

Este método, demostró la ventaja de construir clasificadores no lineales y facilita una división y/o separación efectiva utilizando vectores e hiperplanos, su elección de este último será de acuerdo a la máxima distancia entre las clases denominadas márgenes.

Por lo general los vectores de soporte están en la familia de los algoritmos de clasificación lineal, ya que presentan separadores lineales en el ámbito de características y dimensiones altas, es importante que recordemos que

todo hiperplano en un espacio  $D_i$  – Dimensional,  $\mathfrak{R}D_i$ , se puede representar de manera  $J(x) = \langle w, x \rangle + b$ , donde  $w \in \mathfrak{R}D_i$ , es un vector ortogonal de hiperplano,  $b \in \mathfrak{R}, \langle \cdot, \cdot \rangle$ , este expresa el producto en  $\mathfrak{R}D_i$ . Considerando que este es observado como un clasificador binario sus reglas se denotan de la siguiente manera:  $f(y) = \text{signo}(h(y))$ , donde podemos definir.

$$\text{signo}(y) = \begin{cases} +1, & \text{si } y \geq 0 \\ -1, & y < 0 \end{cases}$$

El SVM, tiene un funcionamiento correlativo de sus datos en un espacio determinado, el mismo que posee características de grandes dimensiones, de manera que los puntos de datos tengan su categorización, incluso si estos no podrían separarse de forma lineal. Se localiza un separador en las categorías que transforman los datos, de forma que dicho separador se pueda extraer como un hiperplano, para un mejor entendimiento se presenta las siguientes ilustraciones.

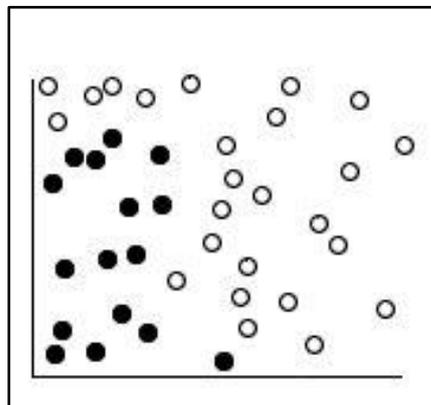


Figura 5. Grupo de datos originales

Fuente: (IBM, 2020)

Para el caso de la ilustración, este puede ser separados por curvas como mostramos en la presente.

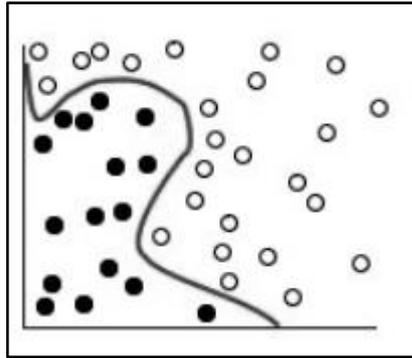


Figura 6. Separación por curvas entre ambas categorías

Fuente: (IBM, 2020)

En la ilustración se evidencia tras la alteración, el límite entre ambas categorías es llamado hiperplano.

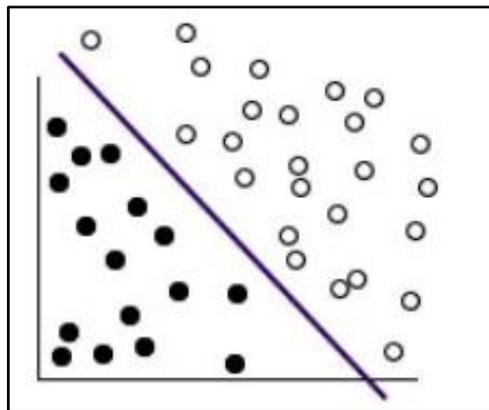


Figura 7. Hiperplano entre ambas categorías

Fuente: (IBM, 2020)

#### 1.4. Formulación del Problema.

¿Qué técnica de clasificación es más eficiente para el diagnóstico depresivo leve?

#### 1.5. Justificación e importancia del estudio.

A nivel de Latinoamérica; la depresión es un problema de gran impacto y necesita la atención necesaria, tanto en su labor social como en las diferentes iniciativas y estrategias, económicas, afortunadamente el gobierno central

brinda y ofrecen herramientas para fortalecer estos casos clínicos y pero no suficientes.

Con esta investigación, se desea reducir un problema evidente en el sector salud y al mismo tiempo sirva para futuras investigaciones sobre el tema.

### **1.6. Hipótesis.**

Las redes neuronales son más eficientes para el diagnóstico depresivo leve.

### **1.7. Objetivos.**

#### **1.7.1. Objetivo general.**

Comparar técnicas de clasificación de aprendizaje de maquina en el diagnóstico de trastorno depresivo leve.

#### **1.7.2. Objetivos específicos.**

- a) Construir un dataset de test de diagnóstico de trastorno depresivo leve
- b) Seleccionar técnica de aprendizaje automático con mejor desempeño
- c) Implementar las técnicas de aprendizaje automático en lenguaje de programación.
- d) Evaluar el desempeño de las técnicas en el diagnóstico del trastorno depresivo leve

## **II. MATERIAL Y MÉTODO**

### **2.1. Tipo y Diseño de Investigación.**

Está fundamentada en la metodología cuantitativa tecnológica, aplicada al diseño cuasi experimental, ya que, mediante técnicas computacionales, estadísticas o matemáticas se podrá demostrar los resultados del presente trabajo de investigación.

### **2.2. Variables, Operacionalización.**

En la opción ítem se espera que se establezca la fórmula de cálculo del indicador.

Tabla 6.

Variables de la investigación

Variables	Indicador	Fórmula	Técnicas
<b>Variable dependiente:</b>  Diagnóstico del trastorno depresivo leve	Precisión	$PR = \frac{VPO}{FPO + VPO}$	Instrumentos electrónicos/ ficha de registro electrónico.
	Exactitud	$E = \frac{VN1 + VPO}{FPO + FNE + VPO}$	
	Recall	$RC = \frac{TP}{FN + TP}$	
	Tiempo de respuesta	$TR = \frac{TF}{TI}$	
<b>Variable Independiente:</b>  Técnicas de Aprendizaje Automático	Consumo de memoria RAM	$RA = \frac{TK - UK}{TK}$	Instrumentos electrónicos/ ficha de registro electrónico.
Consumo de CPU	$CPU = \frac{TCPU}{NCPU}$		

Fuente: Elaboración propia.

## 2.3. Población y muestra.

### 2.3.1. Población

Se consideró una población de diez algoritmos de clasificación los mismos que son los más importantes, (DATAFLAIR TEAM, 2019)

Tabla 7.

Población de técnicas de aprendizaje de máquina

N.º	Técnicas de aprendizaje de máquina
1	Árbol De Decisión
2	K Vecinos Más Cercanos
3	Máquinas Vectores De Soporte

---

4	Naive Bayes
5	Randon Forests
6	Redes Neuronales
7	Regresión Logística
8	Perceptrón
9	C4.5
10	C5.0

---

Fuente: Elaboración propia.

### **2.3.2. Muestra**

Está compuesta por 4 algoritmos con la mayor precisión promedio en estudios enfocados a la depresión, estos fueron seleccionados por conveniencia, para la presente investigación fueron elegidos las técnicas máquinas vectores de soporte, árbol de decisión, redes neuronales, naive bayes, el cuadro comparativo de lo expuesto se encuentra en el anexo 3 tabla 19.

### **2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad.**

Para la recopilación de información se utilizaron ficha de registro electrónico, de acuerdo al siguiente detalle:

- Identificar un hospital para recabar información sobre el diagnóstico de depresión.
- De acuerdo a las fichas de diagnóstico identificar las preguntas de acuerdo a la escala de Zung para su discriminación de la depresión.
- Identificar los puntajes asignadas a cada pregunta.
- Clasificar las equivalencias de la depresión y determinar la categoría a la cual pertenecen.

### **2.5. Procedimiento de análisis de datos.**

En cuando a la evaluación del diagnóstico de la depresión será calculado con base las métricas como precisión, exactitud, recall y para medir las técnicas de

aprendizaje de maquina utilizaremos el consumo de CPU, tiempo de respuesta y consumo de memoria RAM, las fórmulas son las siguientes:

- a. Expresión para evaluar el Tiempo de respuesta

$$RT = FTi - ITi$$

Donde:

RT=tiempo de respuesta

FTi=tiempo final

ITi=tiempo inicial

- b. Expresión para determinar el Consumo de CPU

$$CPU = \frac{TPD\_CPU}{TPI\_CPU}$$

Donde:

CPU = valor resultante de consumo

TPD\_CPU= tiempo en procesar los datos

TPD\_CPI = valor de instrucciones en un tiempo exacto

- c. Expresión para valorar el consumo de memoria ram

$$MRAM = \sum_b^a \frac{CRAM_p}{t}$$

Donde:

MRAM = resultado de consumo

CRAM<sub>p</sub> = valor en el consumo de memoria en ejecución p

t = número de pruebas

## 2.6. Criterios éticos.

### Consistencia

Se realizaron la selección de algoritmos a estudiar, con el objetivo de realizar la comparación y estimar la predicción en el diagnóstico de la depresión

**Validez**

Se validan instrumentos de recopilación de datos y se proponen soluciones a través elección por conveniencia

**Confidencialidad**

Todas las muestras obtenidas para el proceso de la investigación se realizaron en completo estado reservado.

**Derechos del Autor**

En su totalidad todos los materiales de la presente investigación se encuentran debidamente citados y referenciados.

**Objetividad**

El análisis de los contextos se encuentra fundados en los mejores criterios técnicos e imparciales.

**Veracidad**

Teniendo en cuenta la confidencialidad, toda información que se muestra en la presente es complemente en honor a la verdad.

**2.7. Criterios de Rigor Científico.****Consistencia**

El proceso para el análisis resulta de la comparación de algoritmos de estimación para el diagnóstico de la depresión que ser realizara de manera profesional empleando conocimiento de ingeniería para tener una buena precisión de resultado.

**Validez**

Los resultados obtenidos de esta investigación, fueron evaluados de manera correcta para poder tener el mejor producto valido que ayude a resolver el problema planteado.

### **Fiabilidad**

El presente estudio cumple con este principio ya que hace uso de diferentes técnicas y/o algoritmos de aprendizaje automático.

### **Transferibilidad**

Los resultados tienen carácter de ser transferidos a cualquier investigador que siga la misma línea de investigación.

### **Neutralidad**

Según los procesos desarrollados en esta investigación, garantiza la seguridad de que los resultados no pueden ser alterados y/o manipulados por intereses varios.

## **III. RESULTADOS.**

### **3.1. Resultados en Tablas y Figuras.**

Para esta investigación existe el propósito de comparar técnicas de clasificación de aprendizaje de máquina, para determinar cuál es el más adecuado para el diagnóstico de depresión leve.

De acuerdo a la revisión de la literatura, se eligió 4 algoritmos que son más utilizados dentro del grupo de la salud mental, especialmente en el campo de la depresión, los cuales son naive bayes, support vector machine, arboles de decisión y redes neuronales.

El dataset evaluado por estas 4 técnicas consta de 20,070 registros de diagnósticos de depresión clasificados por valores 1 (dentro de lo normal), 2(depresión leve moderada), 3(depresión moderada intensa) y 4 (depresión intensa), de los cuales fueron entrenados y puestos a prueba en porcentajes de 70% y 30% respectivamente, los resultados obtenidos fueron validados por el método de Validación Cruzada, en donde se evaluó los siguientes indicadores : Exactitud , Precisión , Exhaustividad (Recall), Tiempo de ejecución y consumo de memoria RAM, consumo de CPU del algoritmo.

### **Matriz de confusión**

Esta es conocida como matriz de error, representa un resumen que es utilizado para estimar el rendimiento de diversos modelos de clasificación, de acuerdo al conteo se elabora una tabla y se registra el número de predicciones incorrectas y correctas desglosándose por cada clase.

Para entender fácilmente esta matriz recordar que, es positivo cuando la predicción muestra 1 de lo contrario si es cero será negativo. El verdadero o falso hace referencia a la predicción si fuese correcta o no.

El verdadero positivo, nos indica que el valor real y de predicción es positivo el verdadero negativo, resulta cuando al valor real y de prueba es negativo, falso negativo, es considerado cuando el valor real es positivo y negativo el de predicción, falso positivo, resultado real falso y el de predicción verdadero.

#### **d. Exactitud**

Métrica que ayuda a evaluar técnicas y/o algoritmos, esta es representada por la fracción de predicciones que la técnica calculó de manera correcta.

Esta es representada por la fórmula:

$$EX = \frac{VPO + VNE}{VPO + NVE + FPO + FNE}$$

Donde:

VPO: Valores verdaderos positivos

VNE: Valores verdaderos negativos

FPO: Valores falsos positivos

FNE: Valores falsos negativos

A nivel de la estadista este tiene una relación con el sesgo de una predicción. Por otro lado, se indica que para el cálculo se debe realizar la sumatoria de resultados verdaderos positivos (VPO) más verdaderos negativos (VNO) y ser dividido entre los totales de casos (VPO+NVE+FPO+FNE).

En la siguiente figura se ilustra el análisis comparativo de las diferentes técnicas implementadas y su exactitud que estos tienen frente a los datos obtenidos del diagnóstico de depresión.

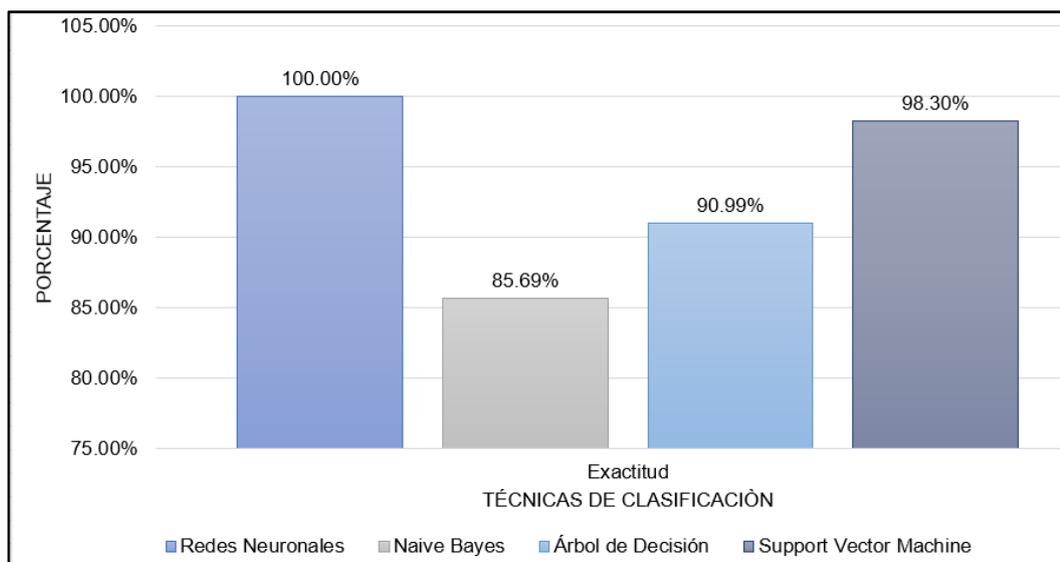


Figura 8. Comparación de exactitud entre técnicas implementadas.

Fuente: elaboración propia.

#### e. Precisión

Para este punto se está evaluando los datos de acuerdo al desempeño de predicciones positivas, y se representa por la siguiente formula.

$$PR = \frac{VPO}{FPO + VPO}$$

Donde:

VPO: datos verdaderos positivos

FPO: datos falsos positivos.

Dicho lo anterior, la precisión es una medida que nos indica que proporción de pacientes a los que diagnosticamos depresión, tienen depresión, para eso se muestra la figura 9, donde se expone los valores alcanzados de acuerdo a los modelos implementados.

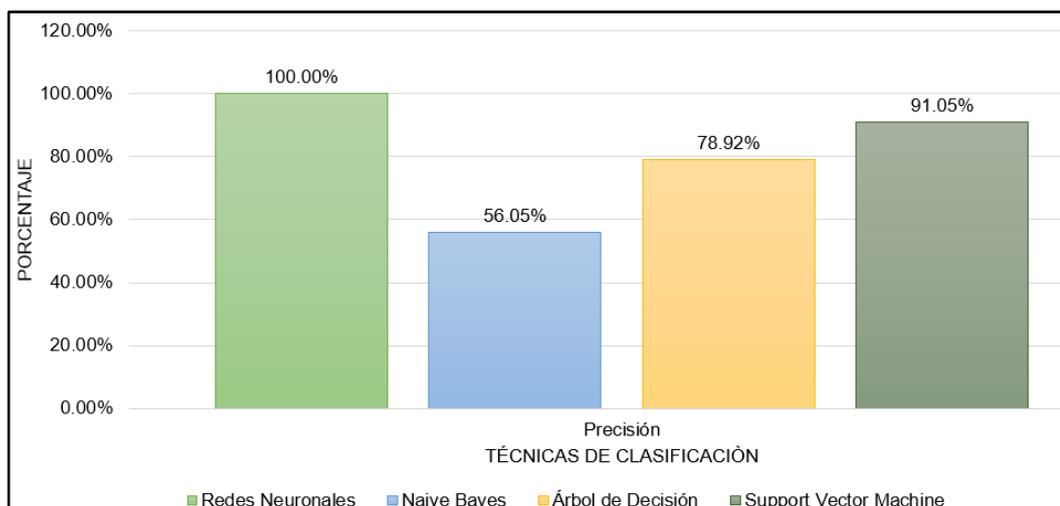


Figura 9. Evaluación de precisión de los algoritmos implementados

Fuente: Elaboración propia.

#### f. Sencibilidad (recall)

Esta métrica trata de medir la proporción de positivos reales que identifico el modelo implementado, está representado por la siguiente formula.

$$RC = \frac{TP}{FN + TP}$$

Donde:

TP: Verdadero positivo

FN: Falso negativo

De acuerdo a los algoritmos implementados, se ha realizado el cálculo con los valores de predicciones correctas, entre los valores totales de positivos, indicándonos la proporción de pacientes que en realidad tuvieron depresión, fue diagnosticado por la técnica como si tuviera depresión, para ello se presenta la figura 10, donde se compara los valores obtenidos por cada técnica.

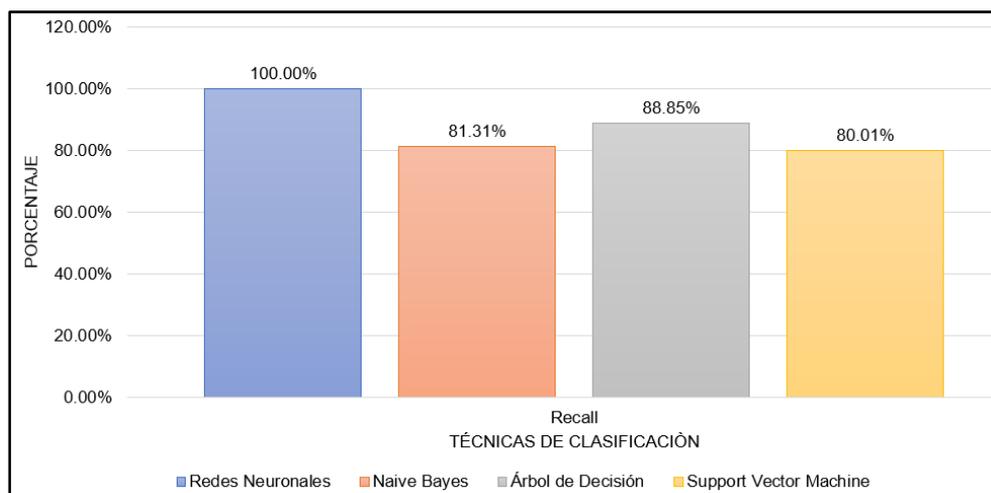


Figura 10. Evaluación de la Recall de las técnicas implementadas  
Fuente: Elaboración propia

Con base a los resultados presentados, se muestra un cuadro comparativo con los valores de cada métrica evaluada, para ello se realizó un muestreo de 5 pruebas, donde se llega a la conclusión que las redes neuronales presentan un 100% de precisión, recall y exactitud respectivamente para el diagnóstico de depresivo leve.

Etiquetas de fila	Exactitud	Precisión	Recall	F1_score
<b>Árbol de Decisión</b>	<b>90.99%</b>	<b>78.92%</b>	<b>88.85%</b>	<b>71.77%</b>
Prueba 01	90.99%	78.92%	88.85%	71.77%
Prueba 02	91.00%	78.92%	88.84%	71.78%
Prueba 03	90.99%	78.92%	88.84%	71.77%
Prueba 04	90.99%	78.92%	88.86%	71.76%
Prueba 05	91.00%	78.92%	88.85%	71.79%
<b>Naive Bayes</b>	<b>85.69%</b>	<b>56.05%</b>	<b>81.31%</b>	<b>65.92%</b>
Prueba 01	85.69%	56.05%	81.31%	65.92%
Prueba 02	85.68%	56.06%	81.31%	65.92%
Prueba 03	85.70%	56.05%	81.30%	65.92%
Prueba 04	85.69%	56.06%	81.30%	65.92%
Prueba 05	85.68%	56.05%	81.31%	65.92%
<b>Redes Neuronales</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>
Prueba 01	100.00%	100.00%	100.00%	100.00%
Prueba 02	100.00%	100.00%	100.00%	100.00%
Prueba 03	100.00%	100.00%	100.00%	100.00%
Prueba 04	100.00%	100.00%	100.00%	100.00%
Prueba 05	100.00%	100.00%	100.00%	100.00%
<b>Support Vector Machine</b>	<b>98.30%</b>	<b>91.05%</b>	<b>80.01%</b>	<b>95.31%</b>
Prueba 01	98.30%	91.05%	80.01%	95.31%
Prueba 02	98.32%	91.03%	80.02%	95.31%
Prueba 03	98.31%	91.05%	80.01%	95.32%
Prueba 04	98.30%	91.06%	80.02%	95.31%
Prueba 05	98.29%	91.05%	80.01%	95.32%
<b>Total general</b>	<b>93.75%</b>	<b>81.51%</b>	<b>87.54%</b>	<b>83.25%</b>

Figura 11. Pruebas de Rendimiento de las técnicas de aprendizaje automático seleccionados. Fuente: Elaboración propia

### g. Tiempo de respuesta

Medir el tiempo de compilación de un algoritmo o programa es un requisito muy importante, ya que desarrollar no solo consiste en crear código, sino que lo desarrollado funcione en un tiempo razonable, para ello se debe saber el tiempo de antes y después de la compilación de código.

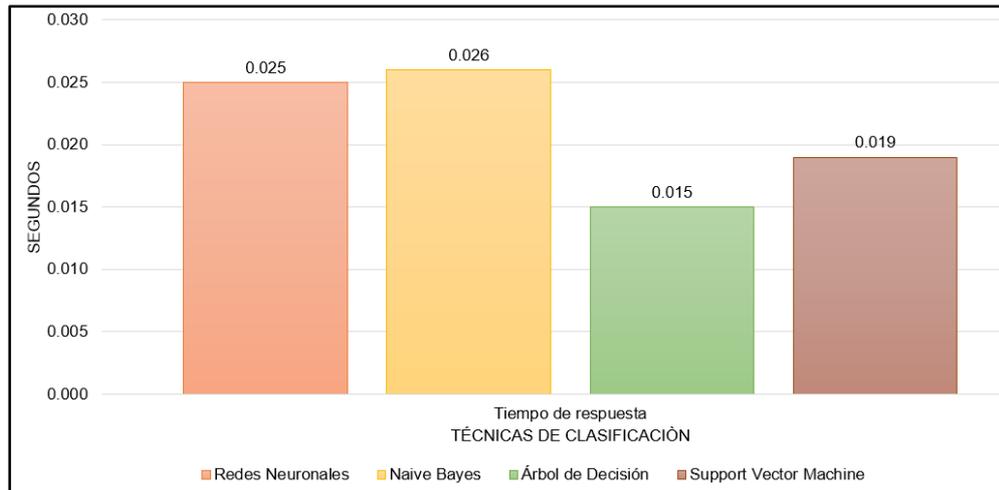
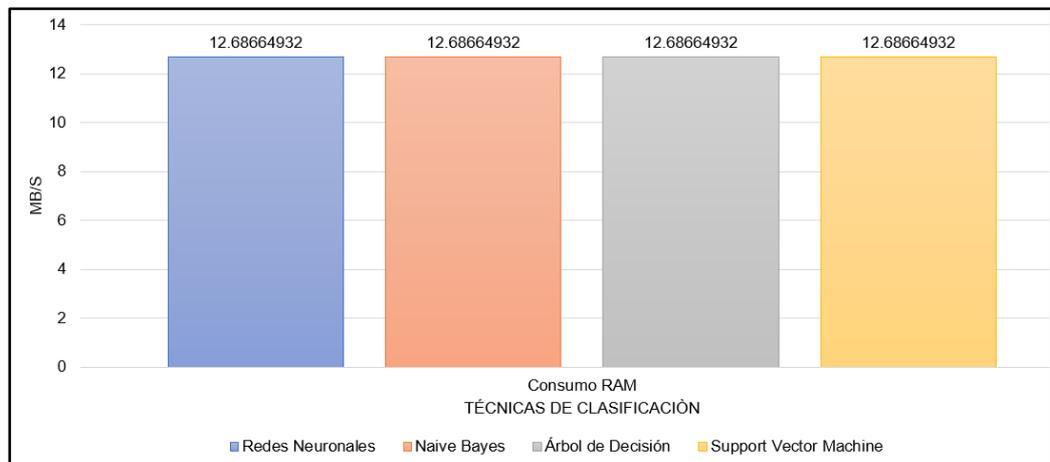


Figura 12. Comparación de tiempo de ejecución de las técnicas implementadas Fuente: Elaboración Propia.

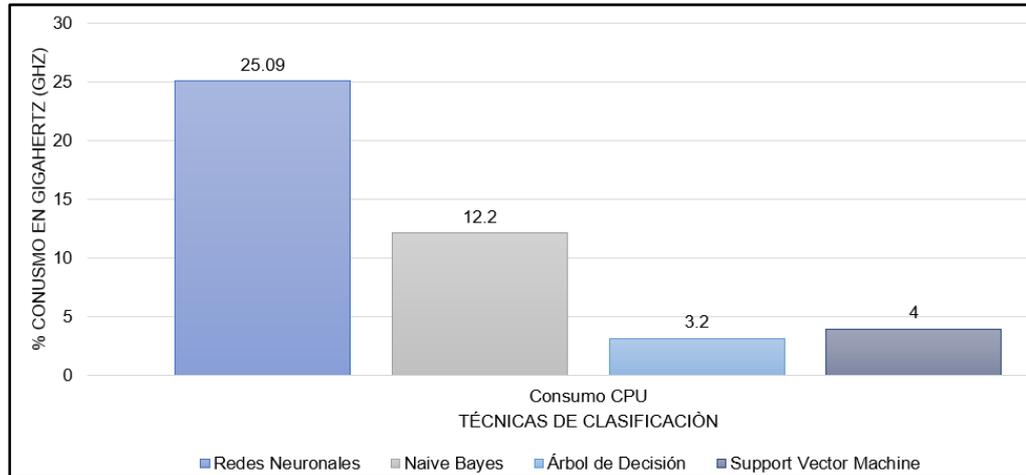
### h. Consumo de memoria ram

Los recursos de un computador son importantes ya que se debe utilizar de manera eficiente, reducir el consumo de memoria es primordial para que el programa realice una ejecución con un rendimiento adecuado, para ello presentamos la comparación de los diferentes algoritmos implementados.



*Figura 13.* Comparación de consumo de memoria de las técnicas implementadas. Fuente: Elaboración Propia.

**i. Consumo de CPU**



*Figura14.* Comparación de Consumo de CPU de todas las técnicas implementadas. Fuente: Elaboración Propia.

De acuerdo a los resultados obtenidos con 5 pruebas, se realizó un cuadro comparativo para determinar cuál de los algoritmos tiene mayor desempeño en el consumo de memoria, consumo de cpu y tiempo de respuesta, queda demostrado que la técnica support vector machine (SVM) muestra valores con el mejor desempeño que los demás.

ALGORITMO DE APRENDIZAJE DE MAQUINA	Tiempo respuesta (En segundos)	Consumo de CPU (%GHz)	Consumo de Memoria (MB)
<b>Árbol de Decisión</b>	<b>0.015</b>	<b>15.77</b>	<b>12.68</b>
Prueba 01	0.015	3.20	12.68
Prueba 02	0.017	12.05	12.69
Prueba 03	0.016	5.50	12.68
Prueba 04	0.014	29.50	12.68
Prueba 05	0.015	28.60	12.68
<b>Naive Bayes</b>	<b>0.026</b>	<b>10.16</b>	<b>12.68</b>
Prueba 01	0.026	12.20	12.67
Prueba 02	0.027	2.00	12.68
Prueba 03	0.026	12.20	12.67
Prueba 04	0.027	12.18	12.68
Prueba 05	0.025	12.20	12.68
<b>Redes Neuronales</b>	<b>0.025</b>	<b>27.32</b>	<b>12.68</b>
Prueba 01	0.025	25.09	12.68
Prueba 02	0.026	31.20	12.68
Prueba 03	0.024	24.00	12.69
Prueba 04	0.025	25.09	12.68
Prueba 05	0.026	31.20	12.68
<b>Support Vector Machine</b>	<b>0.019</b>	<b>15.36</b>	<b>12.68</b>
Prueba 01	0.019	4.00	12.69
Prueba 02	0.020	3.60	12.68
Prueba 03	0.019	15.90	12.68
Prueba 04	0.019	26.40	12.68
Prueba 05	0.020	26.90	12.68
<b>Total general</b>	<b>0.022</b>	<b>17.15</b>	<b>12.68</b>

Figura 15. Pruebas de métricas de los algoritmos seleccionados

Fuente: Elaboración Propia

### 3.2. Discusión de resultados.

Según (Anees Ul, Jamil, Musarrat, Muhammad, & Sungyoung, 2017), en su artículo *“Sentiment analysis of social networking sites (SNS) data using machine learning approach for the measurement of depression”* se ha utilizado teorías de emociones, utilizando técnicas de aprendizaje automático y técnicas naturales de procesamiento del lenguaje en diferentes plataformas de redes sociales, haciendo una comparación entre clasificadores SVM, NB y ME con respecto al análisis del sentimiento del nivel de oración para la medición de la depresión, dicho modelo de votación y la técnica de selección de características ha resuelto tener una precisión de SVM es del 91 %, la precisión de nave bayes es del 83 % y la precisión de la entropía máxima es del 80%, siendo un resultado que está por debajo de nuestro modelo implementado, el cual se ha evaluado a través de un documento técnico llamado “auto escala de Zung”, por intermedio de este se realiza la medición de la depresión a través de información de aspectos

cognitivos, afectivos y somáticos del paciente, dicho documento tiene una especificidad del 63% y sensibilidad del 97%, aprobando un acierto del 82% para la discriminar depresión teniendo respuestas de los paciente de una manera más rápida y acertada, dichas respuesta han sido plasmados en un Dataset y se ha realizado la comparación de los algoritmos resultando una precisión del 100% de la técnica de redes neuronales, con lo expuesto se demuestra que los resultados obtenidos son más eficientes que del estudio en discusión.

Según (Deshpande & Rao, 2018), en su artículo “Depression Detection using Emotion Artificial Intelligence”, nos comenta que aplican el procesamiento del lenguaje natural a las fuentes de Twitter para realizar un análisis de emociones centrado en la depresión, para esto en su método propusieron el uso máquinas de vectores de apoyo y clasificadores Naive-Bayes, generando resultados con una puntuación F1 de 83,29, mientras que SVM ha obtenido una puntuación F1 inferior, de 79,73. La Precisión y la Recuperación siguen la misma tendencia y el Bayes ingenuo multinomial supera al SVM. La precisión del Multinomial Naive Bayes es del 83% y del 79% en el caso de la SVM, sin embargo, en comparación con nuestro modelo los resultados fueron mejores en cuantos a los algoritmos resultando una precisión del 92% para SVM y 91% NB esto es a consecuencia de un mejor manejo de información en cuantos a las características depresivas que tiene cada paciente, así como el entrenamiento y test de datos.

### **3.3. Aporte práctico.**

- a) Construcción del dataset con registros de diagnósticos del trastorno de depresión leve.

Para el desarrollo del presente trabajo, se creó un método para la construcción de un dataset, el cual tendrá información sobre el diagnóstico de esta enfermedad.



*Figura 16.* Método para la construcción del dataset.

Fuente (elaboración propia)

Para la identificación del establecimiento de salud, se ha elegido un hospital del ministerio de salud por la gran demanda de pacientes, que acuden a una cita psicológica para realizar un diagnóstico de depresión de acuerdo a los síntomas que el paciente tiene.

Según los especialistas de este nosocomio, informa que el diagnóstico de este trastorno lo realizan a través de un *documento*, por intermedio de este documento se realiza la medición de la depresión a través de información de aspectos cognitivos, afectivos y somáticos del paciente, dicha documento tiene una especificidad del 63% y sensibilidad del 97%, aprobando un acierto del 82% para la discriminar depresión, así mismo están relacionadas a los cambios autonómicos y sensoriales de los pacientes deprimidos. Para estos casos se han creado preguntas y respuestas sensoriales para realizar el descarte si un paciente tiene o no depresión, para ello se detalla las categorías de la escala de autoevaluación de Zung (Piñerúa Shuhaibar, Estévez, & Suárez Roca, 2020).

Tabla 8.

*Cuestionario del documento técnico de Zung, para discriminar la depresión*

<b>Pregunta</b>	<b>Numero pregunta</b>
¿me siento desalentado, triste, y desmotivado?	(1)
¿tengo llantos continuos o razones para hacerlo?	(2)

---

¿en horas de la mañana me siento bien?	(3)
¿puedes dormir durante la noche?	(4)
¿cómo más que antes?	(5)
¿noto que perdí peso?	(6)
¿gozo cuando, hablo o estoy con varones/mujeres cautivadoras?	(7)
¿me estreñimiento frecuentemente?	(8)
¿mi corazón tiene latidos acelerados?	(9)
¿siento cansancios inoportunos?	(10)
¿tengo intranquilidad sin control?	(11)
¿me resulta fácil hacer, las cosas que hice algún momento?	(12)
¿tengo mi mente clara?	(13)
¿considero tener una vida agusta?	(14)
¿tengo un futuro con esperanzas?	(15)
¿me resulta sencillo tomar mis decisiones?	(16)
¿tengo mal humor siempre?	(17)

---

---

¿deleito las cosas que solía hacer? (18)

¿percibo que soy útil y que me necesitan? (19)

¿si yo muero, estarían mejor? (20)

---

*Nota:* Autoevaluación de Zung (Piñerúa Shuhaibar, Estévez, & Suárez Roca, 2020).

De acuerdo a la escala para la depresión hecha por Zung, se detalló que existe 20 preguntas, cada una de ellas vinculadas a la categoría y síntomas, para su calificación la escala cuenta con 4 tipos de respuestas, “algunas veces”, “muy pocas veces”, “muchas veces” o “casi siempre”, lo que cuenta con puntajes del 1 al 4 respectivamente en el orden descrito, el valor total es comparado en una tabla de equivalencia con un rango de 1 a 100 puntos, este valor obtenido revela el nivel o grado de depresión que tiene el paciente. (Piñerúa Shuhaibar, Estévez, & Suárez Roca, 2020).

Tabla 9.

*Tipos de respuestas y su puntaje*

Opciones de respuesta	Puntaje para cada pregunta
Algunas veces	1
Muy pocas veces	2
Muchas veces	3
Casi siempre	4

*Nota:* Para cada opción de respuesta se asigna un puntaje, los mismos que serán calculados por todas las 20 preguntas descritas en la escala.

Fuente: elaboración propia. (Piñerúa Shuhaibar, Estévez, & Suárez Roca, 2020).

Tabla 10.

*Cuadro de equivalencia de acuerdo al puntaje acumulado*

<b>CATEGORIA</b>	<b>PUNTAJE TOTAL</b>
<b>Dentro de lo normal</b>	<50
<b>Leve moderada</b>	50 - 59
<b>Moderada intensa</b>	60 - 69
<b>Intensa</b>	>70

*Nota:* El puntaje obtenido en las respuestas son comparados con esta tabla de equivalencia para definir los el grado de depresión. Fuente: . (Piñerúa Shuhaibar, Estévez, & Suárez Roca, 2020).

Teniendo las interrogantes y las equivalencias del documento de evaluación, se procedió a realizar la transcripción de todos los diagnósticos hechos a un dataset, considerando los campos más importantes:

Tabla 11.

*Descripción de los atributos a utilizar en el data set*

<b>Nombre de atributos</b>	<b>Descripción del Atributo</b>	<b>Condición</b>
Apellidos y nombres	Apellidos y nombres del paciente	Opcional
Edad	Edad del paciente	Obligatorio
Sexo	Sexo del paciente	Obligatorio
Fecha	Fecha de atención del paciente	Opcional
Dni	Documento de identidad del paciente	Opcional
P_1	¿me siento desalentado, triste, y desmotivado?	Obligatorio
P_2	¿tengo llantos continuos o razones para hacerlo?	Obligatorio
P_3	¿en horas de la mañana me siento bien?	Obligatorio
P_4	¿puedes dormir durante la noche?	Obligatorio

P_5	¿cómo más que antes?	Obligatorio
P_6	¿noto que perdí peso?	Obligatorio
P_7	¿gozo cuando, hablo o estoy con mujeres/hombres cautivadores?	Obligatorio
P_8	¿tengo problemas de constipación?	Obligatorio
P_9	¿mi corazón se acelera más de lo normal?	Obligatorio
P_10	¿siento que me canso sin motivos?	Obligatorio
P_11	¿siento intranquilidad y no lo puedo controlar?	Obligatorio
P_12	¿me resulta fácil hacer, lo que hice algún momento?	Obligatorio
P_13	¿tengo la mente clara como siempre lo he tenido?	Obligatorio
P_14	¿considero tener una vida plena?	Obligatorio
P_15	¿tengo esperanzas en el futuro?	Obligatorio
P_16	¿me resulta fácil tomar mis propias decisiones?	Obligatorio
P_17	¿tengo mal humor siempre?	Obligatorio
P_18	¿deleito las cosas que solía hacer?	Obligatorio
P_19	¿percibo que soy útil y que me necesitan?	Obligatorio
P_20	¿si yo muero, estarían mejor?	Obligatorio
Total	Puntaje acumulado por todas las preguntas	Obligatorio
Categoría	Equivalencia de acuerdo al pago total	Obligatorio

*Nota:* Los atributos descritos son los más importantes para realizar el entrenamiento sobre el diagnóstico de la depresión. . (Piñerúa Shuhaibar, Estévez, & Suárez Roca, 2020).

De acuerdo a los atributos descritos, se ha recolectado un total de 20,073 registros de diagnóstico de trastorno depresivo de pacientes de diversos sexos y edades, el 43% representa el sexo masculino y el 57 % al femenino.

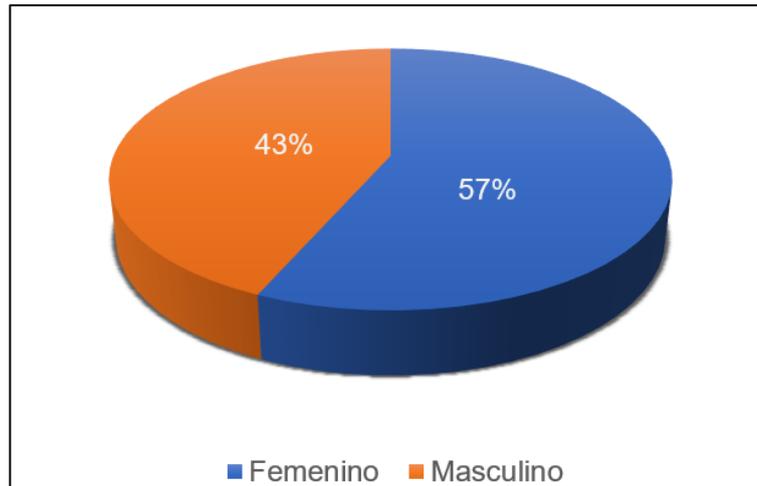


Figura 17. *Registros de dataset por sexo.*

Fuente (elaboración propia)

De los 20,073 registros plasmados en el dataset, es importante resaltar que, según los diagnósticos realizados, el 79.10% representa que el paciente tiene valores normales, 17.45% depresión Intensa, el 0.33% depresión leve moderada y el 3.12% depresión moderada intensa.

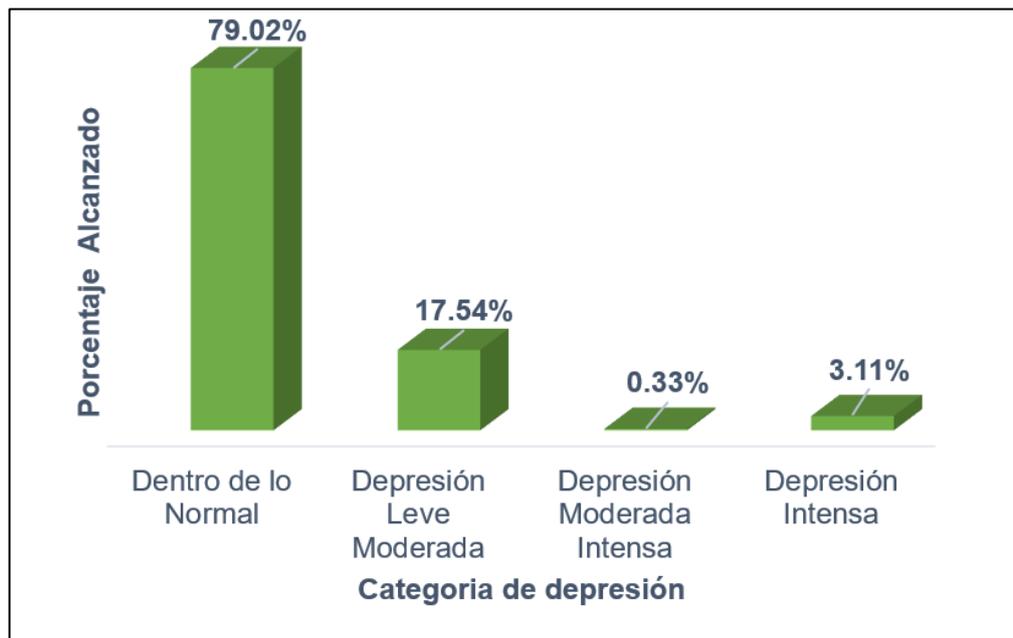


Figura 18. *Tipos de depresión registrados en el dataset.*

Fuente: *Elaboración propia*

- b) Seleccionamos las técnicas de aprendizaje automático con mejor desempeño.

Para la elección seguiremos el siguiente método.



Para la organización de palabras clave, se tomó en cuenta los temas vinculados al estudio del diagnóstico en diferentes enfermedades, la búsqueda se realizó en base de datos científicas Science Direct, IEEE Xplore y Scopus, utilizando las diferentes cadenas de búsqueda.

Tabla 12.

*Cadenas de búsqueda utilizadas en base de datos científicas.*

<b>CADENAS DE BÚSQUEDAS EN BASE DE DATOS CIENTÍFICAS</b>
"classification algorithms" and "review" and "diagnosis"
"classification algorithms" and "review"
"classification algorithms" and "depression" and "diagnosis"
"classification algorithms"

*Nota:* Elaboración propia

Las cadenas de búsqueda nos ayudan a filtrar artículos científicos de acuerdo al tema que se está investigando, la cadena principal que se buscó es "classification algorithms" and "depression" and "diagnosis", teniendo como resultado 86 artículos referentes al tema de diagnóstico de los cuales 10 revistas fueron seleccionadas por que están vinculadas al estudio de la depresión y 76 a otros tipos de diagnóstico.

De todos los artículos encontrados, se realizó una evaluación con base a la precisión promedio, la evaluación de los mismos se encuentra detallados en la tabla 19 del anexo 3.

Tabla 13.

Selección de algoritmos de acuerdo a la precisión promedio

<b>Técnicas de aprendizaje automático</b>	<b>Promedio de precisión</b>
<b>Naive bayes (NB)</b>	<b>83.00%</b>
<b>Redes neuronales artificiales (RNA)</b>	<b>89.92%</b>
<b>Árbol de decisión (DT)</b>	<b>86.67%</b>
<b>Máquinas de vectores de soporte (SVM)</b>	<b>90.49%</b>

Fuente: Elaboración propia

Tabla 14.

Top de las técnicas más usadas para diagnóstico de depresión

Autores	Técnicas				
	Árbol de Decisión (DT)	Naive Bayes (NB)	Support Vector Machine (SVM)	Redes Neuronales (RN)	Otros
(Richter, Fishbain, Fruchter, Richter Levin, & Okon Singer, 2021)	X	X			
(Yalamanchili, Nikhil Sai, Abbaraju, Nadella, & Alluri, 2020)	X		X		X
(Jiang, BinLiu, Wang, Tianyang Liu, & Huanyu Li, 2017)		X	X		X

(Malaquias, y otros, Classification and characterization of children and adolescents with depressive symptomatology using machine learning, 2019)	X			X	X	X
(Hooda, Saxena, Madhulika, & Yadav, 2018)	X	X		X		X
(Anees Ul, Jamil, Musarrat, Muhammad, & Sungyoung, 2017)			X	X		X
(Ji-Won & Kyungyong, 2020)						X
(Deshpande & Rao, 2018)			X	X		
(Allen , y otros, 2017)			X	X		
(Bhakta & Sau, 2016)	X			X		X

*Nota:* Se seleccionó diez artículos con las técnicas de clasificación más usados en el diagnóstico de la depresión, los autores afirman que cada algoritmo cuenta con una precisión aceptable y recomiendan su estudio en futuros tiempos. Fuente: Elaboración propia

**c) Implementación de técnicas de aprendizaje automático en lenguaje de programación.**

El desarrollo de las técnicas de aprendizaje automático fue implementadas a través de la herramienta Google Colaboratory, este entorno de desarrollo nos permite ejecutar código Python usando los servidores de Google, el mismo que no es necesario realizar ninguna configuración, permitiendo crear aplicaciones de manera fácil e interactiva dentro del mundo de la ciencia de datos (Google Colaboratory, 2021).

### **Técnica de aprendizaje Árbol de Decisión (DT)**

Los DT, son algoritmos que son usados tanto para regresión y/o clasificación, consiste principalmente en línea de interrogantes if / else que llevan a un resultado final, Esta técnica modela resultados con alta precisión y son fáciles de interpretar y son usados particularmente en diferentes problemas de la ciencia de datos. (Universidad de Antioquia, 2021).

Los principales procesos de división de nodos en sus atributos nominales se hacen en base decisiones binarias, como por ejemplo usted es varón la contestación puede ser Si o No. Es por ello si uno de sus atributos tiene  $n$  valores posibles, entonces se generarán  $n$  valores particiones binarias entre los valores de  $1 \leq j \leq n$  (Benitez, Escudero, Kanaan, & Masip, 2018).

Para las particiones de atributos numéricos su rango es dividido en dos distancias  $z_i \leq b_c, z_i > b_c$  siendo  $b_c$  (punto medio de corte) una característica  $z_i$  en la evidencia  $E$ .  $b_c$  puede fijarse los índices adecuados para reducir el tamaño del árbol (Benitez, Escudero, Kanaan, & Masip, 2018).

Para la implementación del algoritmo iniciaremos a importar las bibliotecas necesarias, los mismos que se encuentran descritas en el anexo 5, tabla 24.

Se realiza la importación y lectura de datos de un archivo con extensión .CSV, el mismo que contiene diagnósticos de depresión, clasificados por valores 1 (dentro de lo normal), 2 (depresión leve moderada), 3 (depresión moderada intensa) y 4 (depresión intensa), y la como resultado lo valores 0 (sin depresión) y 1 (depresión leve). Ver anexo 5, tabla 24.

Para la división de datos con sus características, se hizo la división de información de variables de ingreso y de salida, así mismo se hace el fraccionamiento entre el grupo de entrenamiento y pruebas, considerándose un 30% de tamaño de pruebas y el 70% para entrenamiento.

Tabla 15.

*División de características y entrenamiento de la técnica árbol de decisión*

---

```
feature_cols = ['P_1','P_2','P_3','P_4','P_5','P_6','P_7','P_8','P_9',  
               'P_10','P_11','P_12','P_13','P_14','P_15','P_16','P_17','P_18',  
               'P_19','P_20']
```

```
Xo = ka[feature_cols]
```

```
yo = ka.RESULTADO
```

```
X_entrenet, X_protest, y_entrenet, y_protest = train_test_split(  
Xo, yo, test_size=0.3, random_state=1)
```

---

Fuente: Elaboración propia

### **Creamos el modelo, entrenamos y optimizamos**

Realizamos el modelo de árbol de decisión usando la librería Scikit-learn, dentro de la implementación se realizó el entrenamiento y además se conseguirá predecir la respuesta de la agrupación de datos de prueba, así mismo se optimizará el modelo y se graficará.

Tabla 16

*Elaboración del árbol de decisión, entrenamiento y predicción.*

---

```
# Crear el objeto clasificador del árbol de decisiones
colof = DecisionTreeClassifier()
#Entrenamiento del árbol de decisión
clfatt = colof.fit(X_entrenet, y_entrenet)
y_pxrd = clfa.predict(X_protest)

dydt_daka = StringIO()
export_graphviz(clfatt, out_file= dydt_daka, filled=True,
rounded=True, special_characters=True, feature_names =
feature_cols, class_names=['1', '0'])
graph = pydotplus.graph_from_dot_data(dydt_daka.getvalue())
graph.write_png('depresion.png')
Image(graph.create_png())
```

---

Fuente: Elaboración propia

### **Evaluación del modelo**

En esta sección se calculó los indicadores de la investigación (ver tabla 6) para luego realizar la comparación de valores predichos y de prueba, los resultados de lo mencionado se encuentran detallada en el anexo 5, tabla 24.

### **Visualización del árbol de decisión**

Para este caso de debe utilizar una función llamada `export_graphviz` de la librería `Scikit -learn`, el cual nos muestra el árbol dentro del ambiente del trabajo de Google Colaboratory. Para trazar el árbol, es necesario la instalación de la librería `graphviz` y `pydotplus`, estos pueden ser instalados usando las sentencias `pip install graphviz` y `pip install pydotplus`, el mismo que al ejecutar crea una imagen visualizable. Ver anexo 5, figura 20.

## Técnica de aprendizaje Naive Bayes (NB)

Es un algoritmo ampliamente usado en machine learning, por su capacidad y eficiencia para el análisis de datos con una gran cantidad de características, Es indispensable mencionar que la técnica estudiada funciona mejor cuando los datos tienen una dimensionalidad alta, el mismo que es capaz de dar un resultado de salida más posible en función a la de entrada. (Manna, 2018).

Este método según la siguiente ecuación:

$P1(c | x)$  en inicio de  $P1(c), P1(x), P1(x | c)$ :

$$q(c | x) = \frac{q(x|c)q(c)}{P1(c)}$$

Entonces:

$q(c | x)$ : La posibilidad de clase posterior (c, objeto) dado el predictor (x, atributos).

$P1(c)$ : La posibilidad previa de clase.

$q(x | c)$ : Posibilidad que es probabilidad del predictor de clase.

$q(x)$ : La posibilidad previa del predictor.

Para el inicio de la implementación en primer lugar se realizó la importación de las librerías, así mismo la carga de datos de acuerdo a las características para la clasificación. Ver anexo 5, tabla 22.

Luego de la lectura del dataset e importación de librerías se realizó el filtrado y vectorización del modelo, el cual inicia con renombrar las columnas del dataset y además seleccionar el porcentaje para entrenamiento y test, que es 70% y 30% respectivamente. Ver anexo 5, tabla 25.

Ya vectorizados y renombrados el modelo, se procedió al entrenamiento de la técnica implementada y luego verificar los resultados del modelo de acuerdo a la implementación realizada, ver anexo 5, tabla 25.

### **Técnica de aprendizaje Redes Neuronales (RNA)**

Este es un algoritmo que está basado en el cerebro humano, el mismo que está conformado por un conjunto de nodos, para cada conjunto de estos son agrupados por neuronas y a estos mismos son asignados un valor y un peso, conectándose entre sí, estas conexiones también poseen su propio valor y a nivel genérico su fórmula principal es la siguiente.

$$B_Q = \sum_{i=1}^n r_i e_i$$

Para cada neurona de ingreso Q, es conectado a varias neuronas de la capa anterior de la red, la sumatoria pondera B de las entras Q es calculado añadido al producto del valor de entrada por cada conexión por r veces el peso de cada conexión e a lo largo de todos los ingresos de n. (Perry, 2018)

En la implementación del algoritmo tiene tres características principales, una de ellas es la conexión entre cada neurona y otras. La siguiente son las señales de entrada que representa la colección de valores numéricos de una capa anterior o una de la capa de entrada, el cual es el único valor normalizado y por último la señal de salida que es un valor numérico que envía a todas las neuronas de la siguiente capa o la de la respuesta si es que esta de salida. (Perry, 2018).

Ya entendiendo el proceso de desarrollo iniciaremos ingresando a la plataforma de google colab, en ellos se importó las librerías necesarias, se leyó el dataset y se empezó a identificar las características de ingreso y variable de salida del dataset. Ver anexo 5, tabla 26.

Para este modelo se utilizó 4 capas una capa de entrada otra de salida y dos ocultas para su entrenamiento, así mismo se está registrando el número de variables de ingreso que es un total de 20 y las interacciones que estos realizarán en un número 3, posterior a ello se realizó compilación y el entrenamiento respectivo.

Tabla 17.

*Creación de capas y entrenamiento de la red neuronal.*

---

```
oculta1 = tensorflow.keras.layers.Dense(units=3, input_shape=[20])
oculta2 = tensorflow.keras.layers.Dense(units=3)
oculta3 = tensorflow.keras.layers.Dense(units=3)
salida = tensorflow.keras.layers.Dense(units=1)
rna = tensorflow.keras.Sequential([oculta1, oculta2, oculta3, salida])
rna.compile(
    optimizer=tensorflow.keras.optimizers.Adam(0.1),
    loss='mean_squared_error'
)
X_entrent, X_protest, y_entrent, y_protest = train_test_split(Xu, yu, test_size=0.30, random_state=0)
print("Comenzando entrenamiento...")
historial = rna.fit(Xu, yu, epochs=100, verbose=False)
print("Modelo entrenado!")
```

---

*Nota:* Elaboración propia.

Del código anterior, se procedió a graficar el historial de entradas y salidas con la librería Matplotlib, Aquí podemos ver como la magnitud de entrenamiento cada vez está siendo reducido y tiende a cero.

*Tabla 18.*

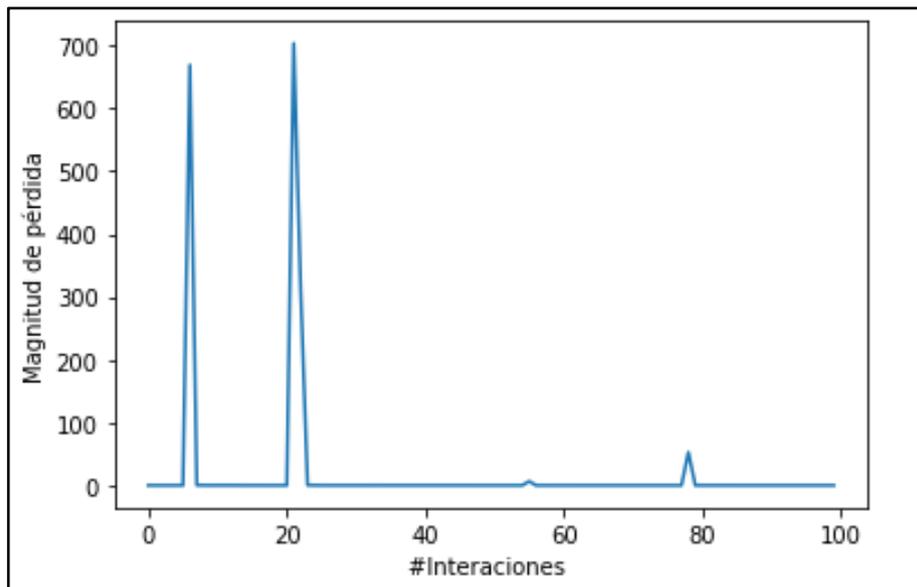
*Graficamos el histograma de la red neuronal indicando su magnitud de perdida.*

---

```
pylt.xlabel("#Interaciones ")  
pylt.ylabel("Magnitud de pérdida")  
pylt.plot(historial.history["loss"])
```

---

Nota: Elaboración propia.



*Figura 19.* Magnitud de la red Neuronal implementada

Fuente: Elaboración Propia.

La evaluación del modelo se detalla en el anexo 5 tabla 26.

## Técnica Support Vector Machine (SVM)

La SVM, es una técnica de aprendizaje supervisado que es utilizada para la regresión, clasificación, fue implementada en los años 90, se desarrolló en clasificación binaria llegando a extenderse a problemas de clasificación de regresión y múltiple, es considerado como uno de los importantes clasificadores a nivel del ámbito de machine learning y aprendizaje estadístico, concentrándose principalmente en el concepto de hiperplano. (Heras, 2020).

El SVM, forman parte de la clase de algoritmos de machine learning que son llamados métodos Kernel, donde estos pueden usar funciones para realizar la transformación de las características, asignando datos a un espacio dimensional, que puede o no ser superior, esto con la finalidad de evaluar su resultado al momento de la separación de clases, su fórmula se representa de la siguiente manera. (Heras, 2020)

$$q(x) = z^t x + b$$

Donde:

z: Son vectores

x: Variable dependiente

b: Variable independiente

conociendo la formula general, la implementación del algoritmo comienza con la importación de las librerías y la lectura del dataset, para este último hacemos una división en variables de entrada (X) y variables de salida (Y), así mismo realizamos el renombre de las columnas y luego se muestra. Ver el anexo 5, tabla 27.

Luego de la lectura de datos, se procedió a la división, del conjunto del mismo tanto para su entrenamiento como para las pruebas, para esto se utilizó la función `train_test_split()`, el mismo que lleva al ingreso de tres parámetros, datos de entrada (X), datos de Salida (Y), estableciéndose 70% de entrenamiento y el 30% de pruebas.

Tabla 17.

*División de datos para las pruebas y entrenamientos*

---

```
>>>from sklearn.model_selection import train_test_split

X_entrent, y_entrent, X_postest, y_postest = train_test_split(Xv,Yv, tes
t_size=0.3,random_state=0)
```

---

Fuente: Elaboración propia

Luego de ello se implementó el modelo Support Vector Machine, con todos los datos leídos y divididos, se realizó el entrenamiento y la evaluación respectivamente.

Tabla 18.

*Implementación y evaluación del algoritmo support vector machine*

---

```
print("--->Comendando el entrenamiento <---")
clf = svm.SVC(decision_function_shape='ovo') # Linear Kernel
clf.fit(Xv,yv)
print("--->Maquina de Soporte Entrenado<---")
```

---

Fuente: Elaboración propia

La evaluación del modelo de acuerdo al código, se detalla en el anexo 5, tabla 27.

#### **d) Evaluación de las técnicas de aprendizaje de máquina.**

Una vez implementado las técnicas, es necesario la evaluación con base a los indicadores tanto para la variable independiente como dependiente, para ello se elaboró una matriz de confusión, que a través de los resultados obtenidos se podrá determinar la evaluación de

métricas como exactitud, precisión, recall, consumo de memoria, tiempo de respuesta y consumo de CPU, en cuando a los resultados por cada algoritmo este se encuentra descrita en el anexo 3 tablas 20 al 23.

## **IV. CONCLUSIONES Y RECOMENDACIONES**

### **4.1. Conclusiones.**

- a) Para seleccionar las técnicas de aprendizaje de máquina, se listo un top de 10 artículos, en ellos se plasmaron las técnicas más usadas en el diagnóstico de la depresión y con base a su precisión promedio fueron elegidos, entre las destacadas están, naive bayes, redes neuronales, support vector machine y arboles de decisión.
- b) El dataset fue construido con un promedio de 20,073 registros de diagnóstico del trastorno depresivo, entre ellos se encuentran depresión leve moderada, moderado intensa, intensa y normal. Luego se seleccionó los atributos necesarios que intervienen en el diagnóstico para el entrenamiento y ejecución de las técnicas de aprendizaje de máquina.
- c) Para la implementación, se usó la plataforma Google Colaboratory, utilizando el lenguaje de programación Python, por la gran variedad de librerías que ayudan en la programación de las técnicas de aprendizaje de máquina, así mismo ofrece los recursos de los servidores de Google ayudando a ejecutar código en el navegador sin configuración alguna y compartiendo código en línea.
- d) Realizada la implementación de las técnicas se ejecutó cinco pruebas teniendo un resultado promedio en su exactitud para los algoritmos arboles de decisión 90.99%, redes neuronales 100%, naive bayes 85.69% y support vector machine 98.30%, lo cual se demuestra que redes neuronales es más eficiente para el diagnóstico de depresión.

## 4.2. Recomendaciones.

- a) Se recomienda usar el método propuesto y su comportamiento en diferentes diagnósticos, estos básicamente enfocados a la salud mental.
- b) Es importante recomendar que la limpieza de datos debe realizarse una vez se tenga identificados los valores necesarios para el ingreso y salida, esto ayudará a que el modelo se procese de una manera más eficiente.
- c) La plataforma recomendable para el análisis e implementación de algoritmos es Google Colaboratory, ya que se ha vuelto un buen aliado dentro del mundo de ciencia de datos, todo se ejecuta online y no es necesario ninguna instalación, además utiliza los recursos propios de los servidores de Google, y su manejo es bien interactivo y comprensible.
- d) La Clasificación internacional de enfermedades (CIE-10), es un manual donde se encuentra clasificados las diversas enfermedades psicológicas, es por ello que se recomienda el uso antes de embarcarse en proyecto orientado a la salud mental u otro tipo de enfermedades.

## REFERENCIAS

- Allen , L., Bryant, L., Daniel , L., Myrna , W., Jonathan, P., Jiook , C., & Shinjae , Y. (2017). Machine learning aided prediction of family history of depression. *New York Scientific Data Summit*, 1-4.
- Anees Ul, H., Jamil, H., Musarrat, H., Muhammad, S., & Sungyoung, L. (2017). Sentiment analysis of social networking sites (SNS) data using machine learning approach for the measurement of depression. *Conferencia internacional de 2017 sobre convergencia de tecnologías de la información y la comunicación (ICTC)* , 138-140.
- Asad, N., Mahmud Pranto, M., Afreen, S., & Islam, M. (2019). Depression Detection by Analyzing Social Media Posts of User. *2019 IEEE International Conference on Signal Processing, Information, Communication and Systems, SPICSCON 2019*, 13-17.
- Baek, J., & Chung, K. (2020). Context Deep Neural Network Model for Predicting Depression Risk Using Multiple Regression. *IEEE Access*, 18171-18181.
- Bellido Riquelme , E., Aragonés Alosilla, L., Bellido Riquelme, E., Castro Paredes, P., & Ramírez Freyre, L. I. (2008). *Guías de Práctica Clínica en Salud Mental y Psiquiatría*.
- Belmonte Fernández, O. (2020). *Introducción al lenguaje de programación Java.: Una guía básica*. Obtenido de Accelerating the world's research: [https://d1wqtxts1xzle7.cloudfront.net/58971457/Introduccion\\_a\\_Java20190420-21890-16e1jjg-with-cover-page-v2.pdf?Expires=1632696768&Signature=LW3ZWheX00EXp-DGyZOUHpM~-cisRLmYdeGqAfTAJTRw9HJvv74y-hkBh7sy1SskdOIMCI45UK2~gNWQCNSuVV-9nuxfyyUMgm18GZ-mGOnXVWt8k](https://d1wqtxts1xzle7.cloudfront.net/58971457/Introduccion_a_Java20190420-21890-16e1jjg-with-cover-page-v2.pdf?Expires=1632696768&Signature=LW3ZWheX00EXp-DGyZOUHpM~-cisRLmYdeGqAfTAJTRw9HJvv74y-hkBh7sy1SskdOIMCI45UK2~gNWQCNSuVV-9nuxfyyUMgm18GZ-mGOnXVWt8k)
- Benitez, R., Escudero, G., Kanaan, S., & Masip, D. (2018). *Inteligencia Artificial Avanzada*. Catalunya: UOC.
- Bhakta, I., & Sau, A. (2016). Prediction of Depression among Senior Citizens using Machine Learning Classifiers. *International Journal of Computer Applications*, 11-16.
- Bhanusree , Y., Nikhil , S., Maruthi Saketh , A., & Sandeep , V. (2020). Real-time Acoustic based Depression Detection using Machine Learning Techniques.

- Conferencia Internacional de Tendencias Emergentes en Ingeniería y Tecnologías de la Información 2020 (ic-ETITE)*, 1-6.
- Biblioteca Estándar de Python. (26 de setiembre de 2021). Obtenido de La biblioteca estándar de Python: <https://docs.python.org/3/tutorial/index.html#tutorial-index>
- Caicedo, E., & López, J. (2017). *Una aproximación práctica a la redes neuronales artificiales*. Santiago de Cali, Colombia: Programa Editorial Universidad del Valle.
- Caicedo, E., & López, J. (2017). *Una aproximación práctica a la redes neuronales artificiales*. Colombia: Universal del Valle.
- Campderrich Falgueras, B. (2013). *Ingeniería del software*. Barcelona: UOC.
- Canos, J., Letelier, P., & Penades, C. (2003). Metodologías Agiles en el Desarrollo de Software. *Taller Metodologías Agiles en el Desarrollo de Software*, 59.
- Ceballos Sierra, J. (2008). *Microsoft C#. Lenguaje de Programación*. México: Alfaomega Grupo Editor.
- Choudhury, A., Khan, M., Nahim, N., Tulon, S., Islam, S., & Chakrabarty, A. (2019). Predicting Depression in Bangladeshi Undergraduates using Machine Learning. *Proceedings of 2019 IEEE Region 10 Symposium, TENSYP 2019*, 789-794.
- Deshpande, M., & Rao, V. (2018). Depression detection using emotion artificial intelligence. *Proceedings of the International Conference on Intelligent Sustainable Systems, ICISS 2017*, 858-862.
- Dirección Regional de Salud Cusco. (2018). *GUÍA DE PRÁCTICA CLÍNICA: EPISODIOS DEPRESIVOS*. Obtenido de [http://www.diresacusco.gob.pe/salud\\_individual/servicios/Gu%C3%ADas%20de%20Pr%C3%A1ctica%20Cl%C3%ADnica%20MINSAPropuestas%20previas%20de%20GPC/Gu%C3%ADas%20Pr%C3%A1cticas%20Cl%C3%ADnicas%20en%20Salud%20Mental/GUIA%20DE%20DEPRESION.pdf](http://www.diresacusco.gob.pe/salud_individual/servicios/Gu%C3%ADas%20de%20Pr%C3%A1ctica%20Cl%C3%ADnica%20MINSAPropuestas%20previas%20de%20GPC/Gu%C3%ADas%20Pr%C3%A1cticas%20Cl%C3%ADnicas%20en%20Salud%20Mental/GUIA%20DE%20DEPRESION.pdf)
- Ferri, C., & Ramírez, M. (2008). *Introducción a la Minería de Datos*. Madrid - España: Pearson Educación.
- Freddy Vásquez. (11 de 08 de 2020). *Sociedad Peruana de Prevención del Suicidio*. Obtenido de <https://peru21.pe/lima/depresion-causa-80-suicidios-peru-atendida-183854-noticia/?ref=p21r>

- García, J., & Pérez, B. (2013). Depresión y los trastornos de ansiedad: autoayuda. *Guías de Autoayuda para la Depresión y los Trastornos de Ansiedad*, 15 - 17.
- Gerych, W., Agu, E., & Rundensteiner, E. (2019). Classifying Depression in Imbalanced Datasets Using an Autoencoder- Based Anomaly Detection Approach. *Proceedings - 13th IEEE International Conference on Semantic Computing, ICSC 2019*, 124 -127.
- González, C., & Valls, A. (2015). Desarrollo de la perspectiva teórica: Revisión de la literatura y construcción del marco teórico. *Opcion*, 984-1001.
- Google Colaboratory. (2021). *Google Colaboratory*. Obtenido de <https://colab.research.google.com/notebooks/welcome.ipynb?hl=es>
- Hassan, A. U., Hussain, J., Hussain, M., Sadiq, M., & Lee , S. (2017). Sentiment analysis of social networking sites (SNS) data using machine learning approach for the measurement of depression. *International Conference on Information and Communication Technology Convergence: ICT Convergence Technologies Leading the Fourth Industrial Revolution, ICTC 2017*, 138-140.
- Heras, J. M. (29 de 08 de 2020). *IArtificial.net*. Obtenido de <https://www.iartificial.net/maquinas-de-vectores-de-soporte-svm/>
- Hooda, M., Saxena, A. R., Madhulika, D., & Yadav, B. (2018). A Study and Comparison of Prediction Algorithms for Depression Detection among Millennials: A Machine Learning Approach. *International Conference on Current Trends in Computer, Electrical, Electronics and Communication, CTCEEC 2017*, 779-783.
- IBM. (julio de 2020). *IBM Knowledge Center*. Obtenido de El modelo de redes neuronales: [https://www.ibm.com/support/knowledgecenter/es/SS3RA7\\_sub/modeler\\_mainhelp\\_client\\_ddita/components/neuralnet/neuralnet\\_model.html](https://www.ibm.com/support/knowledgecenter/es/SS3RA7_sub/modeler_mainhelp_client_ddita/components/neuralnet/neuralnet_model.html)
- Instituto Nacional de Salud Mental. (14 de 06 de 2016). *Radio Programas del Peru*. Obtenido de <https://rpp.pe/vital/expertos/la-depresion-es-el-trastorno-mas-comun-en-el-peru-noticia-971106>

- Jiang, H. H., BinLiu, Z. Y., Wang, L., Tianyang Liu, F. K., & Huanyu Li, X. (2017). Investigation of different speech types and emotions for detecting depression using different classifiers. *Speech Communication*, 39-46.
- Jiang, H., Hu, B., Liu, Z., Yan, L., Wang, T., Liu, F., . . . Li, X. (2020). Investigación de diferentes tipos de discurso y emociones para detectar. *Comunicación oral*, 39 - 46.
- Jimenez, H., Rodriguez, R., & Tiparra, J. (1978). *Diagnóstico de TEA*. Madrid: Latinoamérica SA.
- Jimenez, J. U. (2019). Introducción a R y RStudio. 1-20.
- Ji-Won, B., & Kyungyong, C. (2020). Context Deep Neural Network Model for. *IEEE Access*, 18171-18181.
- John, D., & Raymond, D. (2009). *Introducción a la Programación con JAVA*. Mexico: Mc Graw Hill.
- Joyanes Aguilar, L. (1996). *Fundamentos de Programación - Algoritmos y Estructura de Datos*. Majadahonda - Madrid: Concepción Fernández Madrid.
- Khalil, R., & Al-Jumaily, A. (2017). Machine learning based prediction of depression among type 2 diabetic patients. *Proceedings of the 2017 12th International Conference on Intelligent Systems and Knowledge Engineering*, 1-5.
- Lan, H., & Pan, Y. (2019). A Crowdsourcing Quality Prediction Model Based on. *IEEE*, 314-320.
- Lucio, A., Fernandez, P., & Martines, L. (2013). *Introducción a los Modelos de Regresión*. Mexico: Plaza y Valdes.
- Malaquias, K., Lima, T., Santana, R., Salgado, F., Teodoro, M., & Nobre, C. (2019). Classification and characterization of children and adolescents with depressive symptomatology using machine learning. *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, 534-539.
- Malaquias, K., Lima, T., Santana, R., Salgado, F., Teodoro, M., & Nobre, C. (2019). Classification and characterization of children and adolescents with depressive symptomatology using machine learning. *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, 534-539.

- Mejia, I., Ramirez, R., Jimenez, H., & Rosas, J. (2019). A new method a architecture entreprise. *Conference IEEE bussines*, 200-215.
- Mejia, I., Tuesta, M., & Forero, M. (2020). A new method of enterprise archicture small organizations. *Computer Science Techology*, 150-170.
- Mesenguer, P., & Lopez, R. (2017). *Inteligencia Artificial*. Madrid: SCIC - Consejo Superior de Investigaciones Cientificas.
- Minguillón, J., Casas, J., & Minguillón, J. (2017). *Minería de Datos: Modelos y Algoritmos*. España: UOC.
- Organizacion Mundial de la Salud. (11 de 8 de 2020). Obtenido de <https://peru21.pe/lima/depresion-causa-80-suicidios-peru-atendida-183854-noticia/?ref=p21r>
- Organizacion Mundial de la Salud. (30 de 01 de 2020). Obtenido de <https://www.who.int/es/news-room/fact-sheets/detail/depression>
- Pacheco Patiño, M. A. (2019). *Elementos de Programacion*. Monografía, Lima. Obtenido de <https://repositorio.une.edu.pe/bitstream/handle/UNE/4340/Monografia%20Elementos%20de%20Programaci%c3%b3n.pdf?sequence=1&isAllowed=y>
- Palma, J., & Marín, R. (2008). *Inteligencia Artificial Método, Técnicas y Aplicaciones*. Aravaca - Madrid: Mc Graw Hill.
- Perry, J. S. (14 de 12 de 2018). *IBM & INSTANA*. Obtenido de <https://developer.ibm.com/es/tutorials/cc-artificial-neural-networks-neuroph-machine-learning/>
- Piñerúa Shuhaibar, L., Estévez, J., & Suárez Roca, H. (26 de Setiembre de 2020). *Autescala de Zung para depresión como predictor de la respuesta sensorial y autonómica al dolor*. Obtenido de [http://ve.scielo.org/scielo.php?script=sci\\_arttext&pid=S0535-51332007000400007&lng=es&tlng=es](http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S0535-51332007000400007&lng=es&tlng=es).
- Richter, T., Fishbain, B., Fruchter, E., Richter Levin, G., & Okon Singer, H. (2021). Machine learning-based diagnosis support system for differentiating between clinical anxiety and depression disorders. *Journal of Psychiatric Research*, 199-205.
- Rojas, K. (2018). Identificación de efectos negativos de la TEA en el aprendizaje. *IEEE conference Techology children especial*, 200-215.

- Santana Julio, S., & Mateos Farfan, E. (2014). *El Arte de Programar en R - Un Lenguaje para la Estadística*. México: D.R. © Instituto Mexicano de Tecnología del Agua.
- SZNAJDLEDER, P. (2012). *Java a fondo - estudio del lenguaje y desarrollo de aplicaciones - 2a ed.* México: Alfaomega.
- Tuñoque, L. (10 de 09 de 2020). Obtenido de <https://rpp.pe/vital/salud/salud-mental-la-depresion-el-impacto-de-la-covid-19-y-su-diagnostico-oportuno-noticia-1291532>
- Universidad de Antioquia. (18 de 11 de 2021). *Diplomado de Análisis de datos y Machine Learning en Python*. Obtenido de facultad de ciencias exactas y naturales:  
[https://colab.research.google.com/github/jdamaster/machineLearningDiplomat/blob/master/s10\\_Arboles\\_y\\_clustering.ipynb#scrollTo=2RiEAyMBTo2s](https://colab.research.google.com/github/jdamaster/machineLearningDiplomat/blob/master/s10_Arboles_y_clustering.ipynb#scrollTo=2RiEAyMBTo2s)
- Vargas Cajahuanca, G., Gallegos Cazorla, C. S., Salazar de la Cruz, M. H., Bonilla Untiveros, C. R., & Caballero Ñopo, P. (2019). Guía de práctica clínica basada en evidencias para el tratamiento de depresión en adultos en un hospital especializado en salud mental. Lima, Perú. *Articulos de Facultad de Medicina*, 123 - 130.
- Yalamanchili, B., Kota, N. S., Abbaraju, M. S., Nadella, V., Sai, S., & Alluri, S. V. (2020). Real-time Acoustic based Depression Detection using Machine Learning Techniques. *International Conference on Emerging Trends in Information Technology and Engineering, ic-ETITE 2020*, 1 - 6.
- Yalamanchili, B., Nikhil Sai, k., Abbaraju, M. S., Nadella, V. S., & Alluri, S. V. (2020). Real-time Acoustic based Depression Detection using Machine Learning Techniques. *International Conference on Emerging Trends in Information Technology and Engineering, ic-ETITE 2020*, 1-6.
- Eroglu S., Toprak S., Urgan O, MD, Ozge E. Onur, MD, Arzu Denizbasi, MD, Haldun Akoglu, MD, Cigdem Ozpolat, MD, Ebru Akoglu, Md. 2012. "Guia Practica de Depresión." *Saudi Med J* 33:3–8.
- Ministerio de Salud. 2020. "RD-267-2020-DEPRESION.Pdf."
- Vaishali Narayanrao, Purude, and P. Lalitha Surya Kumari. 2020. "Analysis of Machine Learning Algorithms for Predicting Depression." *2020 International*

*Conference on Computer Science, Engineering and Applications, ICCSEA  
2020. doi: 10.1109/ICCSEA49143.2020.9132963.*

# ANEXOS.

## Anexo 1. Resolución de aprobación del trabajo de investigación



### FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO RESOLUCIÓN N°1000-2021/FIAU-USS

Pimentel, 11 de noviembre de 2021

#### VISTOS:

El Acta de reunión N°0610-2021 del Comité de investigación de la Escuela profesional de INGENIERÍA DE SISTEMAS remitida mediante Oficio N°0359-2021/FIAU-IS-USS de fecha 14 de octubre de 2021, y;

#### CONSIDERANDO:

Que, de conformidad con la Ley Universitaria N° 30220 en su artículo 48° que a letra dice: "La investigación constituye una función esencial y obligatoria de la universidad, que la fomenta y realiza, respondiendo a través de la producción de conocimiento y desarrollo de tecnologías a las necesidades de la sociedad, con especial énfasis en la realidad nacional. Los docentes, estudiantes y graduados participan en la actividad investigadora en su propia institución o en redes de investigación nacional o internacional, creadas por las instituciones universitarias públicas o privadas.";

Que, de conformidad con el Reglamento de grados y títulos en su artículo 21° señala: "Los temas de trabajo de investigación, trabajo académico y *tesis* son *aprobados por el Comité de Investigación* y derivados a la facultad o Escuela de Posgrado, según corresponda, para la emisión de la resolución respectiva. El *periodo de vigencia de los mismos será de dos años*, a partir de su aprobación. En caso un tema perdiera vigencia, el Comité de Investigación evaluará la ampliación de la misma.

Que, de conformidad con el Reglamento de grados y títulos en su artículo 24° señala: La tesis es un estudio que debe denotar rigurosidad metodológica, originalidad, relevancia social, utilidad teórica y/o práctica en el ámbito de la escuela profesional. Para el grado de doctor se requiere una tesis de máxima rigurosidad académica y de carácter original. Es individual para la obtención de un grado; *es individual o en pares para obtener un título profesional*. Asimismo, en su artículo 25° señala: "El tema debe responder a alguna de las líneas de investigación institucionales de la USS S.A.C.".

Que, según documentos de vistos el Comité de investigación de la Escuela profesional de INGENIERÍA DE SISTEMAS acuerda aprobar la modificación de los temas de Tesis a cargo de los estudiantes que se detallan en el anexo de la presente Resolución.

Estando a lo expuesto, y en uso de las atribuciones conferidas y de conformidad con las normas y reglamentos vigentes;

#### SE RESUELVE:

**ARTÍCULO 1°: MODIFICAR**, el tema de la Tesis perteneciente a la línea de investigación de INFRAESTRUCTURA, TECNOLOGÍA Y MEDIO AMBIENTE, a cargo de los estudiantes del Programa de estudios de **INGENIERÍA DE SISTEMAS** según se detalla en el anexo de la presente Resolución.

**ARTÍCULO 2°: MODIFICAR**, la Resolución de Facultad con la que se asigna Asesor especialista y/o Jurado evaluador en el extremo del tema de la tesis quedando tal como se detalla en el anexo de la presente Resolución.

**ARTÍCULO 3°: DEJAR SIN EFECTO**, toda Resolución emitida por la Facultad que se oponga a la presente Resolución.

#### REGÍSTRESE, COMUNÍQUESE Y ARCHÍVESE


Mg. Victor Alanci Tuesta Montesca  
Decano (a) / Facultad De Ingeniería,  
Arquitectura Y Urbanismo  
UNIVERSIDAD SEÑOR DE SIPÁN S.A.C.


MBA. María Noelia Sialer Rivera  
Secretaria Académica / Facultad de Ingeniería,  
Arquitectura y Urbanismo  
UNIVERSIDAD SEÑOR DE SIPÁN S.A.C.

Cc: Interesado, Archivo

**FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO  
RESOLUCIÓN N° 1000-2021/FIAU-USS**

Pimentel, 11 de noviembre de 2021

**ANEXO**

N°	AUTOR(ES)	TEMA DE TESIS ACTUAL	TEMA DE TESIS ANTERIOR	RESOLUCIÓN PREVIA
1	SALAZAR AGUILAR LUIS	DETECCIÓN AUTOMÁTICA DE MENSAJE DE TEXTO OCULTO EN UN ARCHIVO DE AUDIO	EVALUACIÓN DE LOS ALGORITMOS HMM Y DTW PARA MEDIR LA EFICIENCIA EN EL TRÁFICO OCULTO DE VOZ IP	0672-2021/FIAU-USS
2	CARRERA SANCHEZ JOSE ANTONIO	DESARROLLO DE UN MODELO DE CIBERSEGURIDAD BASADO EN EL MARCO NIST V1.1. PARA LA DEFENSA CONTRA ATAQUES CIBERNÉTICOS A UNA UNIVERSIDAD NACIONAL PERUANA	EVALUACIÓN DE MARCOS DE TRABAJO PHP PARA EL DESARROLLO DE APLICACIONES WEB, BAJO LA NORMA ISO/IEC 25010, ENFOCADA A LA CALIDAD EN USO DEL PRODUCTO	0451-2021/FIAU-USS
3	THEOLOGITIS SANCHEZ DIMITRIS IOANNIS	PREDICCIÓN DE LA DEMANDA DE PRODUCTOS PARA PYMES DEDICADAS AL NEGOCIO RETAIL UTILIZANDO REDES NEURONALES ARTIFICIALES	PREDICCIÓN DE LA DEMANDA PARA EL ABASTECIMIENTO DEL INVENTARIO BASADO EN REDES NEURONALES ARTIFICIALES PARA PYMES DEDICADAS AL NEGOCIO RETAIL	0449-2021/FIAU-USS
4	ALFARO YESQUEN LILIANA ELIZABETH	IDENTIFICACIÓN AUTOMÁTICA DE INTENSIDAD DE CLOROFILA EN PLANTAS DE CAPSICUM ANNUUM GROUP MEDIANTE EL PROCESAMIENTO DE IMÁGENES DIGITALES	IDENTIFICACIÓN AUTOMÁTICA DE ESTRÉS HÍDRICO EN PLANTAS DE CAPSICUM ANNUUM GROUP MEDIANTE EL PROCESAMIENTO DE IMÁGENES DIGITALES	1818-2019/FIAU-USS
5	CIEZA CELIS JESUS ABELARDO OJEDA ROMERO ANTHONNY JHONATAN	EVALUACIÓN DEL DESEMPEÑO DE PROTOCOLOS DE SEGURIDAD PARA COMBATIR ATAQUES EN REDES INALÁMBRICAS WI-FI	EVALUACIÓN DEL DESEMPEÑO DE PROTOCOLOS DE SEGURIDAD DE REDES PARA COMBATIR VULNERABILIDADES EN REDES INALÁMBRICAS WI-FI	0700-2021/FIAU-USS
6	TAPIA LLATAS MANUEL AURELIO	COMPARACIÓN DE TÉCNICAS DE CLASIFICACIÓN AUTOMÁTICA PARA LA IDENTIFICACIÓN EFECTIVA DE MALWARE	COMPARACIÓN DE TÉCNICAS DE SISTEMAS INMUNES ARTIFICIALES EN LA IDENTIFICACIÓN DE MALWARE	2320-2020/FIAU
7	CALDERON YNOÑAN PAMELA DEL CARMEN PRIETO NEIRA FRANCK ALBERSON	INTEGRACIÓN DE UN ENFOQUE ÁGIL CON TÉCNICAS DE DISEÑO CENTRADO EN USUARIO (DCU) PARA LA MEJORA DE EXPERIENCIA DE USUARIO (UX)	DESARROLLO DE UN MÉTODO BAJO EL ENFOQUE ÁGIL EN ENTORNOS DE EXPERIENCIA DE USUARIO UI/UX PARA ASEGURAR LA USABILIDAD WEB	0445-2021/FIAU-USS
8	GUEVARA PEREZ ALEX HUMBERTO	COMPARACIÓN DE ALGORITMOS DE SEGMENTACIÓN DE IMÁGENES DIGITALES DE LAS HOJAS DE CAPSICUM ANNUUM ADQUIRIDAS EN AMBIENTE NO CONTROLADO	COMPARACIÓN DE ALGORITMOS DE SEGMENTACION PARA LA DETECCIÓN DE ENFERMEDAD OIDIOPSIS EN AMBIENTES NO CONTROLADOS EN CAPSICUM ANNUUM GROUP MEDIANTE PROCESAMIENTO DE IMÁGENES DIGITALES	1823-2019/FIAU-USS
9	CASTRO MEDINA MIGUEL ANGEL	IMPLEMENTACIÓN DE UN MODELO AD HOC DE GESTIÓN DE SEGURIDAD DE LA	IMPLEMENTACIÓN DE UN MODELO AD HOC DE GESTIÓN DE LA SEGURIDAD	0700-2021/FIAU-USS

**FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO  
RESOLUCIÓN N° 1000-2021/FIAU-USS**

Pimentel, 11 de noviembre de 2021

<b>N°</b>	<b>AUTOR(ES)</b>	<b>TEMA DE TESIS ACTUAL</b>	<b>TEMA DE TESIS ANTERIOR</b>	<b>RESOLUCIÓN PREVIA</b>
		INFORMACIÓN PARA UNA EMPRESA EDITORA DE DIARIO REGIONAL PERUANO	DE LA INFORMACIÓN PARA UNA EMPRESA EDITORA DE DIARIO REGIONAL PERUANO	
10	LUCERO CIEZA ROBINSON	COMPARACIÓN DE TÉCNICAS DE CLASIFICACIÓN DE APRENDIZAJE DE MÁQUINA EN EL DIAGNÓSTICO DEL TRASTORNO DEPRESIVO LEVE	HERRAMIENTA PARA EL DIAGNOSTICO DEL TRASTORNO DEPRESIVO LEVE UTILIZANDO APRENDIZAJE AUTOMATICO	0449-2021/FIAU-USS
11	CARRASCO MARTINEZ LUIS MARLON	EVALUACIÓN DE MÉTODOS DE DETECCIÓN DE INTRUSOS PARA BLOQUEAR Y NOTIFICAR ATAQUES EN LA ZONA DESMILITARIZADA DE LA RED	EVALUACIÓN DE METODOS DE PREVENCIÓN DE INTRUSOS PARA BLOQUEAR Y NOTIFICAR ATAQUES EN LA ZONA DESMILITARIZADA DE LA RED	0447-2021/FIAU-USS



**Anexo 3. Instrumentos de recolección de datos con su respectiva validación de instrumentos.**

**a) Selección de la población de técnicas de aprendizaje de maquina**

Tabla 19.

*Selección de las de técnicas de aprendizaje automático con la mayor precisión promedio.*

<b>N.º</b>	<b>Técnicas de aprendizaje automático</b>	<b>Precisión</b>	<b>Estudio Realizado</b>	<b>Autores</b>	<b>Promedio</b>
1	Árbol de decisión	90.00%	Analysis of Machine Learning Algorithms for Predicting Depression	(Vaishali Narayanrao & Lalitha Surya Kumari 2020)	
2	Árbol de decisión	80.00%	Sentiment analysis of social networking sites (SNS) data using machine learning approach for the measurement of depression	(Hassan, Hussain, Hussain, Sadiq, & Lee , 2017)	86.67%
3	Árbol de decisión	90.00%	Predictors of depression among middle-aged and elderly men and women in Europe: A machine learning approach	(Elizabeth P. & Otros, 2022)	

4	K- vecinos más cercanos	80.00%	Analysis of Machine Learning Algorithms for Predicting Depression	(Vaishali Narayanrao & Lalitha Surya Kumari 2020)	
5	K- vecinos más cercanos	95.44%	Machine learning based prediction of depression among type 2 diabetic patients	(Khalil & Al-Jumaily, 2017)	80.81%
6	K- vecinos más cercanos	67.00%	Predicting Depression in Bangladeshi Undergraduates using Machine Learning	(Choudhury, y otros, 2019)	
7	Naive bayes	83.00%	Sentiment analysis of social networking sites (SNS) data using machine learning approach for the measurement of depression	(Hassan, Hussain, Hussain, Sadiq, & Lee , 2017)	83.00%
8	Naive bayes	83.00%	Depression Detection using Emotion Artificial Intelligence	(Deshpande & Rao, 2018)	
9	Randon Forests	75.00%	Predicting Depression in Bangladeshi Undergraduates using Machine Learning	(Choudhury, y otros, 2019)	75.00%
10	Redes neuronales	93.75%	Machine learning based prediction of depression among type 2 diabetic patients	(Khalil & Al-Jumaily, 2017)	
11	Redes neuronales	80.00%	Context Deep Neural Network Model for Predicting Depression Risk Using Multiple Regression	(Baek & Chung, 2020)	89.92%
12	Redes neuronales	96.00%	Machine Learning Aided Prediction of Family	(Allen , y otros, 2017)	

13	Regresión logística	79.00%	Real-time Acoustic based Depression Detection using Machine Learning Techniques	(Bhanusree , Nikhil , Maruthi Saketh , & Sandeep , 2020)	79.00%
14	Máquinas de vectores de soporte	95.00%	Real-time Acoustic based Depression Detection using Machine Learning Techniques	(Bhanusree , Nikhil , Maruthi Saketh , & Sandeep , 2020)	
15	Máquinas de vectores de soporte	90.00%	Analysis of Machine Learning Algorithms for Predicting Depression	(Vaishali Narayanrao & Lalitha Surya Kumari 2020)	
16	Máquinas de vectores de soporte	100.00%	Depression Detection by Analyzing Social Media Posts of User	(Asad, Mahmud Pranto, Afreen, & Islam, 2019)	
17	Máquinas de vectores de soporte	92.00%	Classifying Depression in Imbalanced Datasets Using an Autoencoder- Based Anomaly Detection Approach	(Gerych, Agu, & Rundensteiner, 2019)	90.49%
18	Máquinas de vectores de soporte	90.00%	Classification and characterization of children and adolescents with depressive symptomatology using machine learning	(Malaquias, y otros, 2019)	
19	Máquinas de vectores de soporte	96.87%	Machine learning based prediction of depression among type 2 diabetic patients	(Khalil & Al-Jumaily, 2017)	

---

20	Máquinas de vectores de soporte	91.00%	Sentiment analysis of social networking sites (SNS) data using machine learning approach for the measurement of depression	(Hassan, Hussain, Hussain, Sadiq, & Lee , 2017)
21	Máquinas de vectores de soporte	79.00%	Depression Detection using Emotion Artificial Intelligence	(Deshpande & Rao, 2018)
22	Máquinas de vectores de soporte	98.00%	Machine Learning Aided Prediction of Family	(Allen , y otros, 2017)
23	Máquinas de vectores de soporte	73.00%	Predicting Depression in Bangladeshi Undergraduates using Machine Learning	(Choudhury, y otros, 2019)

---

b) Para nuestra investigación se utilizó herramientas mecánicas electrónicas, siendo la matriz de confusión un insumo para realizar la comparación de eficiencia de técnicas, se detalla:

Tabla 20.

*Matriz de confusión DT*

		<b>Técnica árbol de decisión</b>	
		Positivos	Negativos
Observación	Positivos	<b>4791</b>	<b>184</b>
	Negativos	<b>358</b>	<b>689</b>

*Nota:* Elaboración propia.

Tabla 21.

*Matriz de confusión NB*

<b>Técnica Naive Bayes</b>			
		Predicción	
		Positivos	Negativos
Observación	Positivos	<b>7212</b>	<b>1089</b>
	Negativos	<b>347</b>	<b>1389</b>

*Nota:* Elaboración propia.

Tabla 22.

*Matriz de confusión RNA*

		<b>Técnica redes neuronales</b>	
		Predicción	
		Positivos	Negativos
Observación	Positivos	<b>4984</b>	<b>0</b>
	Negativos	<b>0</b>	<b>1038</b>

*Nota:* Elaboración propia.

Tabla 23.

*Matriz de confusión SVM*

<b>Técnica support vector machine</b>			
		Predicción	
		Positivos	Negativos
Observación	Positivos	<b>4882</b>	<b>102</b>
	Negativos	<b>0</b>	<b>1038</b>

*Nota:* Elaboración propia.

## Anexo 5. Otros anexos que considere conveniente.

Tabla 24.

*Implementación de la técnica árbol de decisión.*

---

### Código fuente.

---

```
import pandas as kass
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics
import sklearn as sja
from io import StringIO
from IPython.display import Image
from sklearn.tree import export_graphviz
import pydotplus
import time
from sklearn.metrics import classification_report, confusion_matrix, precision_score, f1_score, recall_score, accuracy_score
import psutil

uzl = 'https://raw.githubusercontent.com/Robinsoncieza/datasetdepresion/main/DEPRESION_VAL_OK.csv'
ka = kass.read_csv(uzl, encoding='ISO-8859-1', sep=",")
ka.head()

col_names = ['P_1','P_2','P_3','P_4','P_5','P_6','P_7','P_8','P_9','P_10','P_11','P_12','P_13','P_14','P_15','P_16','P_17','P_18','P_19','P_20','RESULTADO']
feature_cols = ['P_1','P_2','P_3','P_4','P_5','P_6','P_7','P_8','P_9','P_10','P_11','P_12','P_13','P_14','P_15','P_16','P_17','P_18','P_19','P_20']
Xa = ka[feature_cols]
ya = ka.RESULTADO
```

---

---

```
X_entrenet, X_protest, y_entrenet, y_protest = train_test_split(Xa, ya, test_size=0.3, random_state=1)
```

```
# Crear el objeto clasificador del árbol de decisiones
```

```
colof = DecisionTreeClassifier()
```

```
#Entrenamiento del árbol de decisión
```

```
clfa = colof.fit(X_entrenet, y_entrenet)
```

```
y_pxrd = clfa.predict(X_protest)
```

```
dydt_daka = StringIO()
```

```
export_graphviz(clfa, out_file= dydt_daka, filled=True, rounded=True, special_characters=True, feature_names = feature_cols, class_names=['1','0'])
```

```
graph = pydotplus.graph_from_dot_data(dydt_daka.getvalue())
```

```
graph.write_png('depression.png')
```

```
Image(graph.create_png())
```

```
#Evaluación de técnica
```

```
cldor = DecisionTreeClassifier(max_depth=3 , min_samples_split=10).fit(X_entrenet, y_entrenet)
```

```
prrns = cldor.predict(X_protest)
```

```
start = time.time()
```

```
print('=====MÉTRICAS ARBOLES DE DECISION=====')
```

```
print('precision_score◇:', precision_score(y_protest, prrns, average='binary'))
```

```
print('exactitud ◇:', accuracy_score(y_protest, prrns))
```

```
print('Recall◇:', recall_score(y_protest, prrns, average='binary'))
```

```
print('f1_score◇:', f1_score(y_protest, prrns, average='binary'))
```

```
print('Matriz:\n', confusion_matrix(y_protest, prrns))
```

```
fin = time.time()
```

```
print('Consumo de memoria física◇: {}'.format(memory))
```

```
print('Promedio de tiempo de respuesta: {0:5f}seg◇:'.format(fin -start))
```

```
memory = psutil.virtual_memory().total / (1024.0 ** 3)
```

```
print('Grado de consumo de CPU◇:', psutil.cpu_percent(interval=None))
```

---

---

## #Resultados

=====MÉTRICAS ARBOLES DE DECISION=====

precision\_score◊: 0.7892325315005727

exactitud ◊: 0.9099966788442378

Recall◊: 0.6580706781279847

f1\_score◊: 0.7177083333333333

Matriz:

[[4791 184]

[ 358 689]]

Consumo de memoria física◊: 12.686656951904297

Promedio de tiempo de respuesta: 0.015698seg◊:

Grado de consumo de CPU◊: 1.6

---

*Nota:* Elaboración Propia

Tabla 25.

*Implementación de la técnica naive bayes*

---

### **Código fuente.**

---

```
import pandas as kass
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics
import sklearn as sja
import numpy as npy
from sklearn.model_selection import train_test_split
import psutil
from sklearn.metrics import classification_report, confusion_matrix, precision_
score, f1_score, recall_score, accuracy_score
from sklearn.naive_bayes import GaussianNB
import time
```

---

---

```

uzl ='https://raw.githubusercontent.com/Robinsoncieza/datasetdepresion/main/DEPRESION_VAL_OK.csv'
ka = kass.read_csv(uzl, encoding='ISO-8859-1',sep=",")
ka.head()

col_names = ['P_1','P_2','P_3','P_4','P_5','P_6','P_7','P_8','P_9','P_10','P_11','
P_12','P_13','P_14','P_15','P_16','P_17','P_18','P_19','P_20','RESULTADO']
ka.head()

feature_cols = ['P_1','P_2','P_3','P_4','P_5','P_6','P_7','P_8','P_9','P_10','P_11'
,'P_12','P_13','P_14','P_15','P_16','P_17','P_18','P_19','P_20']
Xa = ka[feature_cols]
ya = ka.RESULTADO

X_entrenet, X_protest, y_entrenet, y_protest = train_test_split(Xa, ya, test_size=0.3, random_state=1)

# entreaminado del modelo
nbay = GaussianNB()
nbay.fit(X_entrenet, y_entrenet)
print('Proceso de entrenamiento')
print('Entrenamiento de Naive Bayes--> OK')
# making predictions on the testing set
y_pxrd = clfa.predict(X_protest)

##Evaluacion de la tecnica.
X_entrenet, X_protest, y_entrenet, y_protest = train_test_split(Xa, ya, test_size=0.5, random_state=0)
cnbay = GaussianNB()
prtns = cnbay.fit(X_entrenet, y_entrenet).predict(X_protest)
start = time.time()

```

---

---

```

print('=====MÉTRICAS NAIVE BAYES=====
')
print('precision_score◇:', precision_score(y_protest,prtns, average='binary'))
print('exactitud ◇:', accuracy_score(y_protest, prtns))
print('Recall◇:',recall_score(y_protest,prtns, average='binary'))
print('f1_score◇:',f1_score(y_protest, prtns,average='binary'))
print('Matriz:\n',confusion_matrix(y_protest, prtns))
fin = time.time()
print('Consumo de memoria física◇: {}'.format(memory))
print('Promedio de tiempo de respuesta: {0:5f}seg◇:'.format(fin -start))
memory = psutil.virtual_memory().total / (1024.0 ** 3)
print('Grado de consumo de CPU◇:',psutil.cpu_percent(interval=None))

#Resultados.
=====MÉTRICAS NAIVE BAYES=====

precision_score◇: 0.5605326876513317
exactitud ◇: 0.856929361362957
Recall◇: 0.8001152073732719
f1_score◇: 0.6592311343141908
Matriz:
[[7212 1089]
 [ 347 1389]]
Consumo de memoria física◇: 12.686656951904297
Promedio de tiempo de respuesta: 0.023690seg◇:
Grado de consumo de CPU◇: 1.8

```

---

*Nota:* Elaboración Propia

Tabla 26.

*Implementación de técnica de redes neuronales*

---

**Código fuente.**

---

```
import pandas as kass
import numpy as npy
import tensorflow as tenf
import psutil
from sklearn.metrics import classification_report, confusion_matrix, precision_score, f1_score, recall_score, accuracy_score
from sklearn.neural_network import MLPClassifier
import time
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt

uzl = 'https://raw.githubusercontent.com/Robinsoncieza/datasetdepresion/main/DEPRESION_VAL_OK.csv'
dts = kass.read_csv(uzl, encoding='ISO-8859-1', sep=",")

entrada = dts.iloc[:, 2:22]
salida = dts.iloc[:, 23:24]

X = entrada.to_numpy()
y = salida.to_numpy()

#print(dts.shape)
#dts.head()
#print(salida.columns)

dts.columns = ['EDAD', 'SEXO', 'P_1', 'P_2', 'P_3', 'P_4', 'P_5', 'P_6', 'P_7', 'P_8', 'P_9', 'P_10', 'P_11', 'P_12', 'P_13', 'P_14', 'P_15', 'P_16', 'P_17', 'P_18', 'P_19', 'P_20', 'TOTAL', 'RESULTADO']
```

---

---

```

feature_cols = ['P_1','P_2','P_3','P_4','P_5','P_6','P_7','P_8','P_9','P_10','P_11','P
_12','P_13','P_14','P_15','P_16','P_17','P_18','P_19','P_20']
Xu = dts[feature_cols] # Características
yu = dts.RESULTADO # Variable objetivo

#diseño de capas
oculta1 = tenf.keras.layers.Dense(units=3, input_shape=[20])
oculta2 = tenf.keras.layers.Dense(units=3)
oculta3 = tenf.keras.layers.Dense(units=3)
salida = tenf.keras.layers.Dense(units=1)
rna = tenf.keras.Sequential([oculta1, oculta2,oculta3,salida])

rna.compile(
    optimizer=tenf.keras.optimizers.Adam(0.1),
    loss='mean_squared_error'
)

X_entrent, X_protest, y_entrent, y_protest = train_test_split(Xu, yu, test_size=0.
30, random_state=0)
print("Comenzando entrenamiento...")
historial = rna.fit(Xu, yu, epochs=100, verbose=False)
print("Modelo entrenado!")

pylt.xlabel("#Interacciones ")
pylt.ylabel("Magnitud de pérdida")
pylt.plot(historial.history["loss"])

#Evaluacion de Técnica
crna =MLPClassifier(random_state=1,max_iter=300).fit(X_entrent, y_entrent)
prtns = crna.predict(X_protest)

start = time.time()

```

---

---

```

print('=====MÉTRICAS REDES NEURONALES==
=====')

print('precision_score◇:', precision_score(y_protest,prtns, average='binary'))
print('exactitud ◇:', accuracy_score(y_protest, prtns))
print('Recall◇:',recall_score(y_protest,prtns, average='binary'))
print('f1_score◇:',f1_score(y_protest, prtns,average='binary'))
print('Matriz:\n',confusion_matrix(y_protest, prtns))
fin = time.time()
print('Consumo de memoria física◇: {}'.format(memory))
print('Promedio de tiempo de respuesta: {0:5f}seg◇:'.format(fin -start))
memory = psutil.virtual_memory().total / (1024.0 ** 3)
print('Grado de consumo de CPU◇:',psutil.cpu_percent(interval=None))

```

#Resultados

```

=====MÉTRICAS REDES NEURONALES=====
precision_score◇: 1.0
exactitud ◇: 1.0
Recall◇: 1.0
f1_score◇: 1.0
Matriz:
[[4984  0]
 [  0 1038]]
Consumo de memoria física◇: 12.686656951904297
Promedio de tiempo de respuesta: 0.026482seg◇:
Grado de consumo de CPU◇: 8.9

```

---

*Nota:* Elaboración Propia

Tabla 27.

*Implementación de técnica support vector machine*

---

**Código fuente.**

---

```
%matplotlib inline
import numpy as npy
import matplotlib.pyplot as pipol
import seaborn as sebr; sebr.set()
import pandas as kass
from sklearn.metrics import classification_report, confusion_matrix, precision_score, f1_score, recall_score, accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn import svm
import time
from sklearn.pipeline import make_pipeline
from sklearn.svm import LinearSVC
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn import svm

uzl = 'https://raw.githubusercontent.com/Robinsoncieza/datasetdepresion/main/DEPRESION_VAL_OK.csv'
dts = kass.read_csv(uzl, encoding='ISO-8859-1', sep=",")

dts.columns = ['EDAD', 'SEXO', 'P_1', 'P_2', 'P_3', 'P_4', 'P_5', 'P_6', 'P_7', 'P_8', 'P_9', 'P_10', 'P_11', 'P_12', 'P_13', 'P_14', 'P_15', 'P_16', 'P_17', 'P_18', 'P_19', 'P_20', 'TOTAL', 'RESULTADO']
feature_cols = ['P_1', 'P_2', 'P_3', 'P_4', 'P_5', 'P_6', 'P_7', 'P_8', 'P_9', 'P_10', 'P_11', 'P_12', 'P_13', 'P_14', 'P_15', 'P_16', 'P_17', 'P_18', 'P_19', 'P_20']
Xv = dts[feature_cols] # Características
yv = dts.RESULTADO # Variable objetivo
```

---

---

```

dts.shape

print(X)

print(y)
X_ent, X_post, y_ent,y_post = train_test_split(Xv, yv, test_size=0.3, random_state=0)
print("--->Comendando el entrenamiento <---")

clf = make_pipeline(StandardScaler(),
LinearSVC(random_state=0, tol=1e-5))

clf.fit(Xv, yv)

print("--->Maquina de Soporte Entrenado<---")

#Evaluación de técnica

clasfd= SVC(class_weight ='balanced', C=5).fit(X_ent, y_ent)
prtns = clasfd.predict(X_post)
start = time.time()
print('=====MAQUINAS VECTORES DE SOPORTE====')
print('precision_score: ', precision_score(y_protest,prtns, average='binary'))
print('exactitud : ', accuracy_score(y_protest, prtns))
print('Recall: ',recall_score(y_protest,prtns, average='binary'))
print('f1_score: ',f1_score(y_protest, prtns,average='binary'))
print('Matriz:\n',confusion_matrix(y_protest, prtns))
fin = time.time()
print('Consumo de memoria física: {}'.format(memory))
print('Promedio de tiempo de respuesta: {0:5f}seg:'.format(fin -start))
memory = psutil.virtual_memory().total / (1024.0 ** 3)
print('Grado de consumo de CPU: ',psutil.cpu_percent(interval=None))

```

---

---

#Resultados

=====MAQUINAS VECTORES DE SOPORTE=====

exactitud: 0.9830621056127532

Recall: 1.0

precision\_score: 0.9105263157894737

f1\_score: 0.9531680440771351

Matriz:

[[4882 102]

[ 0 1038]]

Promedio de tiempo de respuesta: 0.013248seg

Grado de consumo de CPU: 5.5

Consumo de memoria física: 12.686656951904297

---

*Nota:* Elaboración Propia

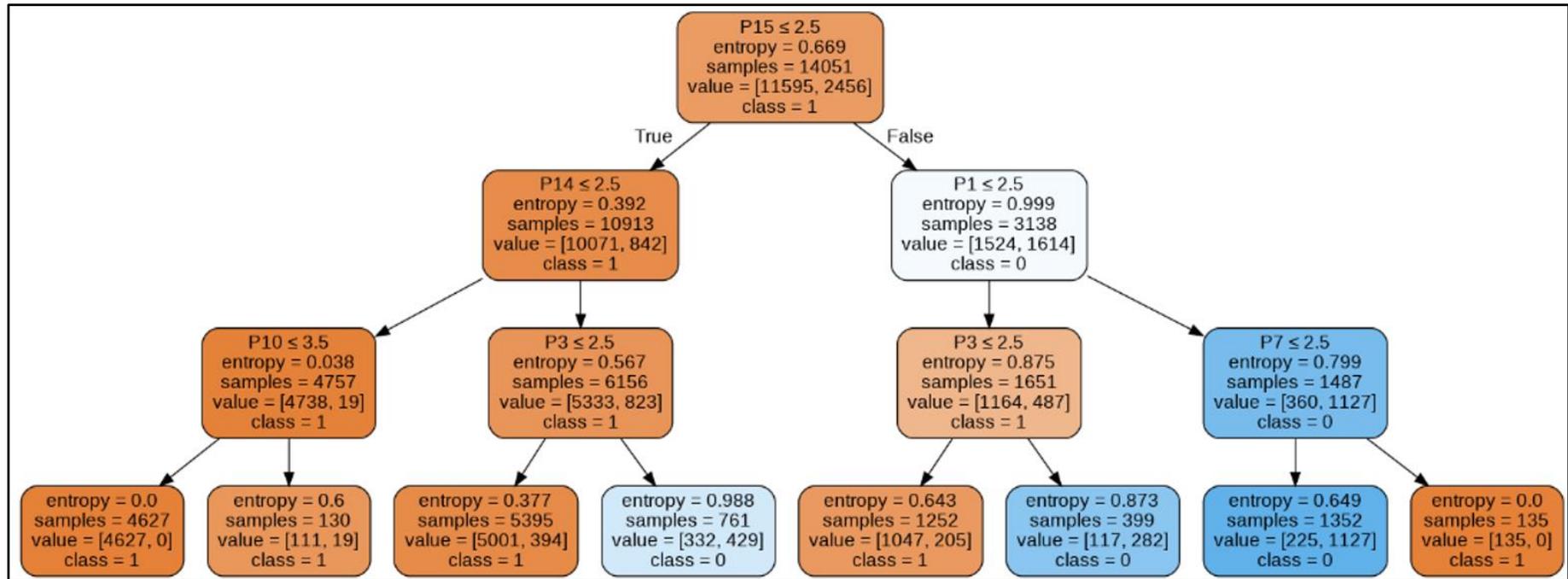


Figura 20. Árbol de decisión Implementado. Fuente: Elaboración propia

TÉCNICAS DE APRENDIZAJE DE MAQUINA - DEPRESION LEVE ☆

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda [Se han guardado todos los cambios](#)

+ Código + Texto

RAM Disco

Comentario Compartir Editar

### IMPLEMENTACIÓN DE TÉCNICA ÁRBOL DE DECISIÓN

```
[1] import pandas as kass
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics
import sklearn as sja
```

```
[2] url = 'https://raw.githubusercontent.com/Robinsoncieza/datasetdepression/main/DEPRESION_VAL_OK.csv'
ka = kass.read_csv(url, encoding='ISO-8859-1', sep=",")
ka.head()
```

	EDAD	SEXO	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_10	P_11	P_12	P_13	P_14	P_15	P_16	P_17	P_18	P_19	P_20	TOTALES	RESULTADO
0	66	1	1	2	1	1	2	3	1	2	1	2	3	4	1	2	1	2	3	4	1	4	41	0
1	26	1	2	1	4	2	1	3	2	3	2	3	4	4	3	4	3	4	4	4	1	3	57	1
2	31	1	2	1	2	3	4	1	2	3	1	2	3	4	1	2	1	2	3	4	1	2	44	0
3	23	0	1	2	3	1	4	3	1	1	1	2	4	3	1	2	1	3	4	4	1	1	43	0
4	48	1	2	1	2	1	4	3	1	2	1	2	4	1	2	2	1	2	3	4	1	2	41	0

```
[11] col_names = ['P_1','P_2','P_3','P_4','P_5','P_6','P_7','P_8','P_9','P_10','P_11','P_12','P_13','P_14','P_15','P_16','P_17','P_18','P_19','P_20','RESULTADO']
feature_cols = ['P_1','P_2','P_3','P_4','P_5','P_6','P_7','P_8','P_9','P_10','P_11','P_12','P_13','P_14','P_15','P_16','P_17','P_18','P_19','P_20']
Xa = ka[feature_cols]
ya = ka.RESULTADO

X_entrenet, X_protest, y_entrenet, y_protest = train_test_split(Xa, ya, test_size=0.3, random_state=1)
```

```
[4] # Crear el objeto clasificador del árbol de decisiones
colof = DecisionTreeClassifier()
#Entrenamiento del árbol de decisión
clfa = colof.fit(X_entrenet, y_entrenet)
y_pxrd = clfa.predict(X_protest)
```

0 s completado a las 21:46

Figura 21. Implementación de técnica de árbol de decisión. Fuente: Elaboración propia

TÉCNICAS DE APRENDIZAJE DE MAQUINA - DEPRESION LEVE ☆

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Guardando...

+ Código + Texto

```
[5] from io import StringIO
from IPython.display import Image
from sklearn.tree import export_graphviz
import pydotplus

dydt_daka = StringIO()
export_graphviz(clfa, out_file= dydt_daka, filled=True, rounded=True, special_characters=True, feature_cols= feature_cols, class_names=['1', '0'])
graph = pydotplus.graph_from_dot_data(dydt_daka.getvalue())
graph.write_png('depression.png')
Image(graph.create_png())
```

0 s completado a las 21:46

Figura 22. Impresión de Árbol de decisión creado. Fuente: Elaboración propia

```
from sklearn.tree import DecisionTreeClassifier
import time
from sklearn.metrics import classification_report, confusion_matrix, precision_score, f1_score, recall_score, accuracy_score
import psutil

#Evaluación de técnica

cldor = DecisionTreeClassifier(max_depth=3, min_samples_split=10).fit(X_entrenet,y_entrenet)
prtns = cldor.predict(X_protest)
start = time.time()
print('=====MÉTRICAS ARBOLES DE DESICION=====')
print('precision_score@:', precision_score(y_protest,prtns, average='binary'))
print('exactitud @:', accuracy_score(y_protest, prtns))
print('Recall@:', recall_score(y_protest,prtns, average='binary'))
print('f1_score@:', f1_score(y_protest, prtns, average='binary'))
print('Matriz:\n',confusion_matrix(y_protest, prtns))
fin = time.time()
print('Consumo de memoria física@: {}'.format(memory))
print('Promedio de tiempo de respuesta: {0:5f}seg@:'.format(fin -start))
memory = psutil.virtual_memory().total / (1024.0 ** 3)
print('Grado de consumo de CPU@:',psutil.cpu_percent(interval=None))
```

```
=====MÉTRICAS ARBOLES DE DESICION=====
precision_score@: 0.7892325315005727
exactitud @: 0.9099966788442378
Recall@: 0.6580706781279847
f1_score@: 0.7177083333333333
Matriz:
[[4791 184]
 [ 358 689]]
Consumo de memoria física@: 12.686656951904297
Promedio de tiempo de respuesta: 0.015698seg@:
Grado de consumo de CPU@: 1.6
```

[ ]

0 s completado a las 21:46

Figura 23. Evaluación de técnica árbol de decisión. Fuente: Elaboración propia

TÉCNICAS DE APRENDIZAJE DE MAQUINA - DEPRESION LEVE ☆

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda [Se han guardado todos los cambios](#)

+ Código + Texto

RAM Disco

Comentario Compartir Editar

### Implementación de Algoritmo Naive Bayes

```
[12] import pandas as kass
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics
import sklearn as sja
import numpy as npy

uzl = 'https://raw.githubusercontent.com/Robinsoncieza/datasetdepression/main/DEPRESION_VAL_OK.csv'
ka = kass.read_csv(uzl, encoding='ISO-8859-1', sep=",")
ka.head()

col_names = ['P_1', 'P_2', 'P_3', 'P_4', 'P_5', 'P_6', 'P_7', 'P_8', 'P_9', 'P_10', 'P_11', 'P_12', 'P_13', 'P_14', 'P_15', 'P_16', 'P_17', 'P_18', 'P_19', 'P_20', 'RESULTADO']
ka.head()
```

	EDAD	SEXO	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_10	P_11	P_12	P_13	P_14	P_15	P_16	P_17	P_18	P_19	P_20	TOTALES	RESULTADO
0	66	1	1	2	1	1	2	3	1	2	1	2	3	4	1	2	1	2	3	4	1	4	41	0
1	26	1	2	1	4	2	1	3	2	3	2	3	4	4	3	4	3	4	4	4	1	3	57	1
2	31	1	2	1	2	3	4	1	2	3	1	2	3	4	1	2	1	2	3	4	1	2	44	0
3	23	0	1	2	3	1	4	3	1	1	1	2	4	3	1	2	1	3	4	4	1	1	43	0
4	48	1	2	1	2	1	4	3	1	2	1	2	4	1	2	2	1	2	3	4	1	2	41	0

```
[14] from sklearn.model_selection import train_test_split

feature_cols = ['P_1', 'P_2', 'P_3', 'P_4', 'P_5', 'P_6', 'P_7', 'P_8', 'P_9', 'P_10', 'P_11', 'P_12', 'P_13', 'P_14', 'P_15', 'P_16', 'P_17', 'P_18', 'P_19', 'P_20']
Xa = ka[feature_cols]
ya = ka.RESULTADO

X_entrenet, X_protest, y_entrenet, y_protest = train_test_split(Xa, ya, test_size=0.3, random_state=1)
```

```
[16] # entrenado del modelo
nbay = GaussianNB()
nbay.fit(X_entrenet, y_entrenet)
print('Proceso de entrenamiento')
print('Entrenamiento de Naive Bayes--> OK')
# making predictions on the testing set
y_pxd = clfa.predict(X_protest)
```

0 s completado a las 21:46

Figura 24. Lectura de datos de técnica Neive Bayes. Fuente: Elaboración propia

```
# making predictions on the testing set
y_pxr = c1fa.predict(X_protest)

Proceso de entrenamiento
Entrenamiento de Naive Bayes--> OK

import psutil
from sklearn.metrics import classification_report, confusion_matrix, precision_score, f1_score, recall_score, accuracy_score
from sklearn.naive_bayes import GaussianNB
import time
X_entrenet, X_protest, y_entrenet, y_protest = train_test_split(Xa, ya, test_size=0.5, random_state=0)
cnbay = GaussianNB()
prtns = cnbay.fit(X_entrenet, y_entrenet).predict(X_protest)
start = time.time()
print('=====MÉTRICAS NAIVE BAYES=====')
print('precision_score@: ', precision_score(y_protest, prtns, average='binary'))
print('exactitud @: ', accuracy_score(y_protest, prtns))
print('Recall@: ', recall_score(y_protest, prtns, average='binary'))
print('f1_score@: ', f1_score(y_protest, prtns, average='binary'))
print('Matriz:\n', confusion_matrix(y_protest, prtns))
fin = time.time()
print('Consumo de memoria física@: {}'.format(memory))
print('Promedio de tiempo de respuesta: {0:5f}seg@:'.format(fin - start))
memory = psutil.virtual_memory().total / (1024.0 ** 3)
print('Grado de consumo de CPU@: ', psutil.cpu_percent(interval=None))

=====MÉTRICAS NAIVE BAYES=====
precision_score@: 0.5605326876513317
exactitud @: 0.856929361362957
Recall@: 0.8001152073732719
f1_score@: 0.6592311343141908
Matriz:
[[7212 1089]
 [ 347 1389]]
Consumo de memoria física@: 12.686656951904297
Promedio de tiempo de respuesta: 0.023690seg@:
Grado de consumo de CPU@: 1.8

[ ]
```

0 s completado a las 21:46

Figura 25. Entrenamiento y evaluación de técnica Naive Bayes. Fuente: Elaboración propia

TÉCNICAS DE APRENDIZAJE DE MAQUINA - DEPRESION LEVE ☆

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda [Se han guardado todos los cambios](#)

+ Código + Texto

RAM Disco Editar

### Implementación técnica Redes Neuronales

```
[00] import pandas as kass
import numpy as npy
import tensorflow as tenf
import psutil
from sklearn.metrics import classification_report, confusion_matrix, precision_score, f1_score, recall_score, accuracy_score
from sklearn.neural_network import MLPClassifier
import time
from sklearn.model_selection import train_test_split

uzl = 'https://raw.githubusercontent.com/Robinsoncieza/datasetdepression/main/DEPRESION_VAL_OK.csv'
dts = kass.read_csv(uzl, encoding='ISO-8859-1', sep=",")

entrada = dts.iloc[:, 2:22]
salida = dts.iloc[:, 23:24]

X = entrada.to_numpy()
y = salida.to_numpy()

#print(dts.shape)
#dts.head()
#print(salida.columns)

dts.columns = ['EDAD', 'SEXO', 'P_1', 'P_2', 'P_3', 'P_4', 'P_5', 'P_6', 'P_7', 'P_8', 'P_9', 'P_10', 'P_11', 'P_12', 'P_13', 'P_14', 'P_15', 'P_16', 'P_17', 'P_18', 'P_19', 'P_20', 'TOTALES', 'RESULTADO']
feature_cols = ['P_1', 'P_2', 'P_3', 'P_4', 'P_5', 'P_6', 'P_7', 'P_8', 'P_9', 'P_10', 'P_11', 'P_12', 'P_13', 'P_14', 'P_15', 'P_16', 'P_17', 'P_18', 'P_19', 'P_20']
Xu = dts[feature_cols] # Características
yu = dts.RESULTADO # Variable objetivo

#diseño de capas
oculta1 = tenf.keras.layers.Dense(units=3, input_shape=[20])
oculta2 = tenf.keras.layers.Dense(units=3)
oculta3 = tenf.keras.layers.Dense(units=3)
salida = tenf.keras.layers.Dense(units=1)
rna = tenf.keras.Sequential([oculta1, oculta2, oculta3, salida])

rna.compile(
    optimizer=tenf.keras.optimizers.Adam(0.1),
    loss='mean_squared_error'
)
```

0 s completado a las 21:46

Figura 26. Lectura de datos de técnica de Redes Neuronales. Fuente: Elaboración propia

TÉCNICAS DE APRENDIZAJE DE MAQUINA - DEPRESION LEVE ☆

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda [Se han guardado todos los cambios](#)

+ Código + Texto

RAM Disco

Comentario Compartir Editar

```
[82] X_entrent, X_protest, y_entrent, y_protest = train_test_split(Xu, yu, test_size=0.30, random_state=0)
print("Comenzando entrenamiento...")
historial = rna.fit(Xu, yu, epochs=100, verbose=False)
print("Modelo entrenado!")
```

Comenzando entrenamiento...  
Modelo entrenado!

```
[83] import matplotlib.pyplot as plt
plt.xlabel("#Interacciones ")
plt.ylabel("Magnitud de pérdida")
plt.plot(historial.history["loss"])
```

[<matplotlib.lines.Line2D at 0x7f2d0f998f90>]

```
[84] #Evaluacion de Técnica
crna =MLPClassifier(random_state=1,max_iter=300).fit(X_entrent, y_entrent)
prtns = crna.predict(X_protest)

start = time.time()
print('-----MÉTRICAS REDES NEURONALES-----')
print('precision_score@:', precision_score(y_protest,prtns, average='binary'))
print('exactitud @:', accuracy_score(y_protest, prtns))
print('Recall@:',recall_score(y_protest,prtns, average='binary'))
print('f1_score@:',f1_score(y_protest, prtns,average='binary'))
print('Matriz:\n',confusion_matrix(y_protest, prtns))
fin = time.time()
print('Consumo de memoria física@: {}'.format(memory))
print('Promedio de tiempo de respuesta: {0:5f}seg@:'.format(fin -start))
memory = psutil.virtual_memory().total / (1024.0 ** 3)
print('Grado de consumo de CPU@:',psutil.cpu_percent(interval=None))
```

0 s completado a las 21:46

Figura 27. Entrenamiento y Evaluación de técnica redes neuronales. Fuente: Elaboración propia

```
#Evaluación de Técnica
crna =MLPClassifier(random_state=1,max_iter=300).fit(X_entrent, y_entrent)
prtns = crna.predict(X_protest)

start = time.time()
print('=====MÉTRICAS REDES NEURONALES=====')

print('precision_score: ', precision_score(y_protest,prtns, average='binary'))
print('exactitud @:', accuracy_score(y_protest, prtns))
print('Recall@:',recall_score(y_protest,prtns, average='binary'))
print('f1_score@:',f1_score(y_protest, prtns,average='binary'))
print('Matriz:\n',confusion_matrix(y_protest, prtns))
fin = time.time()
print('Consumo de memoria física@: {}'.format(memory))
print('Promedio de tiempo de respuesta: {0:5f}seg@:'.format(fin -start))
memory = psutil.virtual_memory().total / (1024.0 ** 3)
print('Grado de consumo de CPU@:',psutil.cpu_percent(interval=None))

=====MÉTRICAS REDES NEURONALES=====
precision_score@: 1.0
exactitud @: 1.0
Recall@: 1.0
f1_score@: 1.0
Matriz:
[[4984  0]
 [ 0 1038]]
Consumo de memoria física@: 12.686656951904297
Promedio de tiempo de respuesta: 0.026482seg@:
Grado de consumo de CPU@: 8.9
```

Figura 28. Resultados de evaluación de técnica redes neuronales. Fuente: Elaboración propia

TÉCNICAS DE APRENDIZAJE DE MAQUINA - DEPRESION LEVE ☆

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda [Se han guardado todos los cambios](#)

+ Código + Texto

RAM Disco

Comentario Compartir Editar

### Implementación Técnica Support Vector Machine

```

%matplotlib inline
import numpy as npy
import matplotlib.pyplot as pipol
import seaborn as sns; sns.set()
import pandas as kass
from sklearn.metrics import classification_report, confusion_matrix, precision_score, f1_score, recall_score, accuracy_score

u21 = 'https://raw.githubusercontent.com/Robinsoncieza/datasetdepresion/main/DEPRESION_VAL_OK.csv'
dts = kass.read_csv(u21, encoding="ISO-8859-1", sep=",")

dts.columns = ['EDAD', 'SEXO', 'P_1', 'P_2', 'P_3', 'P_4', 'P_5', 'P_6', 'P_7', 'P_8', 'P_9', 'P_10', 'P_11', 'P_12', 'P_13', 'P_14', 'P_15', 'P_16', 'P_17', 'P_18', 'P_19', 'P_20', 'TOTALES', 'RESULTADO']
feature_cols = ['P_1', 'P_2', 'P_3', 'P_4', 'P_5', 'P_6', 'P_7', 'P_8', 'P_9', 'P_10', 'P_11', 'P_12', 'P_13', 'P_14', 'P_15', 'P_16', 'P_17', 'P_18', 'P_19', 'P_20']
Xv = dts[feature_cols] # Características
yv = dts.RESULTADO # Variable objetivo

dts.shape

print(X)

print(y)

```

	P_1	P_2	P_3	P_4	P_5	P_6	...	P_15	P_16	P_17	P_18	P_19	P_20
0	1	2	1	1	2	3	...	1	2	3	4	1	4
1	2	1	4	2	1	3	...	3	4	4	4	1	3
2	2	1	2	3	4	1	...	1	2	3	4	1	2
3	1	2	3	1	4	3	...	1	3	4	4	1	1
4	2	1	2	1	4	3	...	1	2	3	4	1	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...
20068	3	2	2	2	3	2	...	2	2	2	2	1	2
20069	1	2	1	1	2	1	...	1	3	4	1	1	4
20070	1	1	1	2	1	1	...	1	1	4	4	1	4
20071	1	2	2	1	1	2	...	1	4	4	4	2	2
20072	2	2	4	2	1	4	...	4	4	2	4	2	2

```

[20073 rows x 20 columns]
0      0
1      1
2      0
3      0
4      0
.
20068  0
20069  0
20070  0
20071  0
20072  0
Name: RESULTADO, Length: 20073, dtype: int64

```

0 s completado a las 21:46

Figura 29. Lectura de datos de técnica Support Vector Machine. Fuente: Elaboración propia

```
from sklearn.model_selection import train_test_split

X_ent, X_post, y_ent, y_post = train_test_split(Xv, yv, test_size=0.3, random_state=0)

from sklearn.pipeline import make_pipeline
from sklearn.svm import LinearSVC
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn import svm

print("---->Comendando el entrenamiento <---")

clf = make_pipeline(StandardScaler(),
LinearSVC(random_state=0, tol=1e-5))

clf.fit(Xv, yv)

print("---->Maquina de Soporte Entrenado<---")

---->Comendando el entrenamiento <---
---->Maquina de Soporte Entrenado<---
/usr/local/lib/python3.7/dist-packages/sklearn/svm/_base.py:1208: ConvergenceWarning: Liblinear failed to converge, increase the number of iterations.
ConvergenceWarning,
```

Figura 30. Entrenamiento de técnica Support Vector Machine. Fuente: Elaboración propia

```
import time
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, f1_score, precision_score, recall_score, classification_report, confusion_matrix
import psutil
#Evaluación de técnica
clasificador = SVC(class_weight='balanced', C=5).fit(X_train,y_train)
predictions = clasificador.predict(X_test)
start = time.time()
#print('predictions:',len(predictions))
#print(y)
print('=====MAQUINAS VECTORES DE SOPORTE=====')
print('exactitud:', accuracy_score(y_test,predictions))
print('Recall:',recall_score(y_test,predictions, average='binary'))
print('precision_score:', precision_score(y_test,predictions, average='binary'))
print('f1_score:',f1_score(y_test,predictions,average='binary'))
print('Matriz: \n',confusion_matrix(y_test,predictions))
fin = time.time()

print('Promedio de tiempo de respuesta: {0:5f}seg'.format(fin -start))
memory = psutil.virtual_memory().total / (1024.0 ** 3)
print('Grado de consumo de CPU:',psutil.cpu_percent())
print('Consumo de memoria física: {}'.format(memory))

=====MAQUINAS VECTORES DE SOPORTE=====
exactitud: 0.9830621056127532
Recall: 1.0
precision_score: 0.9105263157894737
f1_score: 0.9531680440771351
Matriz:
[[4882 102]
 [ 0 1038]]
Promedio de tiempo de respuesta: 0.013248seg
Grado de consumo de CPU: 5.5
Consumo de memoria física: 12.686656951904297
```

Figura 31. Resultados de técnica Support Vector Machine. Fuente: Elaboración propia