Predicción de actividad humana en teléfonos inteligentes para la oportuna localización de sobrevivientes en catástrofes

Andrés Camilo Caraballo Álvarez

Universidad Antonio Nariño
Facultad de Ingeniería de Sistemas
Bogotá, Colombia
2021

Predicción de actividad humana en teléfonos inteligentes para la oportuna localización de sobrevivientes en catástrofes

Andrés Camilo Caraballo Álvarez

Trabajo de grado como requisito para optar al título de:
Ingeniero de Sistemas y computación

Director:
Juan Camilo Ramírez

Asesora Metodológica:
Rosalba Cruz Cepeda
Licenciada en Educación

Tipo de Proyecto:
Proyecto de investigación

Universidad Antonio Nariño
Facultad de Ingeniería de Sistemas
Bogotá, Colombia
2021

AGRADECIMIENTOS

A mi mamá que, aunque se presentaran adversidades siempre me guío paso a paso y tuvo la paciencia que la caracteriza para poder formarme tanto moral como profesionalmente, es gracias a ella que aquí puedo demostrar la persistencia, vocación y demás virtudes que me enseñó desde el comienzo de mi vida.

A la universidad Antonio Nariño que me permitió y me brindó todas las herramientas para formarme como profesional de manera idónea y de alta calidad.

A mi familia la cual siempre tuvo fe en mí, y que a pesar de lo que ocurriese siempre conté con el apoyo de ellos.

A aquellos compañeros y amigos que a lo largo de mi paso por la universidad me acompañaron en este proceso, que me dieron apoyo en momentos difíciles de la carrera.

Por ultimo agradecer al profesor Juan Camilo Ramírez quien con paciencia y esmero me apoyó en todo el proceso de realización del proyecto e incentivó mi gusto por esta área de la ingeniería.

CONTENIDO

LISTA DE IMÁGENES

Pág.

LISTA DE TABLAS

RESUMEN

Actualmente los sistemas de comunicación móvil dependen de la infraestructura física (torres de telecomunicaciones, por ejemplo), con el fin de mantener su calidad. Esto permite la interacción entre cualquier persona sin importar la distancia entre ellas; sin embargo, nadie está exento de estar expuesto a un posible desastre natural. En general, un desastre natural impediría la funcionalidad de comunicación que los dispositivos móviles permiten en circunstancias normales debido al daño causado a la infraestructura física. En muchos desastres ocurridos en los últimos años, las comunicaciones quedan bloqueadas por daños de diversa índole, imposibilitando y dificultando el encuentro de posibles sobrevivientes en momentos donde el tiempo es crucial. Actualmente, el grupo de investigación LACSER de la Universidad Antonio Nariño se encuentra desarrollando el proyecto de investigación titulado “Sistema de comunicación para sobrevivientes de un desastre basado en una red ad hoc de teléfonos”, cuyo objetivo es desarrollar una aplicación móvil llamada Conecta2 utilizando redes ad hoc para establecer comunicación entre sobrevivientes en una situación de desastre por medio dispositivos móviles en ausencia de las redes convencionales de celular e internet. Este trabajo de grado contempló la evaluación de varios modelos de aprendizaje automático para predecir automáticamente si un teléfono inteligente está siendo utilizado en determinado momento por un ser humano a partir de las lecturas de los sensores del dispositivo. Se pretende que estos modelos le permitan a la aplicación Conecta2 identificar aquellos teléfonos inteligentes que en una situación de desastre se encuentren realmente en manos de un ser humano, por ejemplo, un sobreviviente; de tal forma que la comunicación dada entre dispositivos que posean la aplicación se pueda priorizar para economizar recursos, como la batería. Los modelos diseñados en este trabajo de grado fueron entrenados utilizando un conjunto de datos que el proyecto de investigación recopiló y fueron implementados utilizando diferentes técnicas de clasificación para luego ser evaluados a través de diferentes métricas, incluyendo la curva ROC y el F1. Esta evaluación demuestra que los modelos pueden apoyar la localización de sobrevivientes en una situación de desastre y por lo tanto ser incorporados posteriormente a la aplicación Conecta2 para tal propósito.

Abstract

Currently mobile communication systems depend on the physical infrastructure (telecommunications towers, for example), in order to maintain their quality. This allows interaction between anyone regardless of the distance between them, however, no one is exempt from being exposed to a possible natural disaster. In general, a natural disaster would impede the communication functionality that mobile devices allow under normal circumstances due to damage done to the physical infrastructure. In many disasters that have occurred in recent years, communications are blocked by various kinds of damage, making it impossible and difficult to find possible survivors at times when time is of the essence. Currently the LACSER research group of the Antonio Nariño University is developing the research project entitled "Communication system for survivors of a disaster based on an ad hoc network of telephones", whose objective is to develop a mobile application called Conecta2 using ad hoc networks. hoc to establish communication between survivors in a disaster situation through mobile devices in the absence of conventional cellular and internet networks. This degree work contemplated the evaluation of several machine learning models to automatically predict whether a smartphone is being used by a human being at a certain moment from the readings of the device's sensors. These models are intended to allow the Conecta2 application to identify those smartphones that in a disaster situation are actually in the hands of a human being, for example, a survivor; in such a way that the communication given between devices that have the application can be prioritized to save resources, such as the battery. The models designed in this degree work were trained using a set of data that the research project collected and were implemented using different classification techniques to later be evaluated through different metrics, including the ROC curve and the F1. This evaluation shows that the models can support the location of survivors in a disaster situation and therefore be later incorporated into the Conecta2 application for this purpose.

INTRODUCCIÓN

Los avances que ha tenido la comunicación en teléfonos móviles en los últimos años han sido amplios, garantizando la comunicación sin importar las distancias, por medio de infraestructuras diseñadas para tener un funcionamiento óptimo en cuanto a las redes de los dispositivos móviles (Cuenca, 2012). Sin embargo, en caso de algún desastre natural estos dispositivos perderían esta capacidad debido a los daños ocasionados a la infraestructura de red de la que dependen.

Como se evidenció en casos puntuales como los de Chile y Japón, cada uno sufriendo catástrofes naturales en los años 2010 y 2011 respectivamente, se evidenciaron las fallas y daños graves que tuvieron sus redes de comunicación por un tiempo prolongado, por lo cual muchas vidas se perdieron (Hormazábal, 2019; McClelland, 2011).

Acorde a la necesidad de tener una comunicación oportuna en dichas circunstancias, la Universidad Antonio Nariño se encuentra en el desarrollo del proyecto de investigación llamado “Sistema de comunicación para sobrevivientes de un desastre basado en una red ad hoc de teléfonos inteligentes”, el cual tiene como objetivo el desarrollo de una aplicación móvil de mensajería, llamada Conecta2, la cual permita la comunicación entre dispositivos aún cuando la infraestructura de red no esté disponible, como suele ocurrir luego de un desastre natural. Como parte del proyecto se busca que Conecta2 pueda priorizar la comunicación con aquellos dispositivos que cuenten con la aplicación Conecta2 y que estén siendo usados por personas en el momento, ya que pueden ser sobrevivientes. Para lograr esto último, en el marco de este trabajo de grado se propuso utilizar varios métodos de aprendizaje automático para predecir automáticamente si un teléfono inteligente está siendo utilizado en determinado momento por un ser humano a partir de las lecturas de los sensores del dispositivo.

Más puntualmente, este trabajo de grado consiste en el entrenamiento y evaluación de varios modelos de aprendizaje automático para identificar automáticamente si un teléfono inteligente se encuentra en manos de un ser humano, haciendo la predicción a partir de los sensores de los dispositivos. La recopilación del conjunto

de datos base se hizo por parte del proyecto de investigación, utilizando una aplicación móvil llamada *SensorReader* que fue desarrollada en otro trabajo de grado adscrito al mismo proyecto de investigación.

En este trabajo de grado se utilizaron varias técnicas de clasificación, tales como máquinas de vectores de soporte y redes neuronales. Estos modelos serán incorporados posteriormente a Conecta2, con el fin que pueda diferenciar entre si un ser humano se encuentra manipulando un dispositivo inteligente, o, por el contrario, si el ser humano NO se encuentra manipulando el dispositivo.

La detección de actividad humana es un proceso que representa un reto importante para los avances tecnológicos actuales, debido a que la información de donde se extrae, o donde se busca predecir, es en la mayoría de ocasiones poco clara, es decir, estos no brindan la información explicita del comportamiento humano, esto se debe hacer por medio de análisis previos; por ejemplo, en el caso de la lectura de sensores en dispositivos inteligentes, se debe comparar la información que regularmente se recolecta en forma numérica, frente a las posibles acciones de los individuos en estudio, de esta manera poder realizar un análisis completo.

Este documento está estructurado de la siguiente manera: el capítulo 1 contemplará todo lo relacionado al planteamiento del problema, su justificación, objetivos y alcance; el capítulo 2 hablará acerca del marco teórico necesario para la comprensión general del problema planteado, junto a este el marco legal correspondiente a la realización del mismo, y por último el estado del arte donde se contemplan proyectos similares a este; el capítulo 3 correspondiente al marco metodológico se regirá bajo la premisa de los pasos a seguir para dar solución al problema planteado de manera óptima y clara; el capítulo 4 describirá los resultados obtenidos al realizar el procedimiento mencionado en el capítulo anterior, brindando una visión general del estudio; Por último, el capítulo 5 brindará una serie de conclusiones y recomendaciones frente al entregable final y un análisis breve de lo realizado.

1. PLANTEAMIENTO DEL PROBLEMA

1.1. DESCRIPCIÓN DEL PROBLEMA

Actualmente las redes de comunicación en dispositivos móviles se encuentran ligadas a infraestructura física, la cual hace más amplia su cobertura; sin embargo, esta infraestructura no está exenta de sufrir daños considerables como consecuencia de que ocurra una catástrofe natural, lo que impediría el uso de todas las funcionalidades de comunicación que un dispositivo móvil permitiría en circunstancias normales.

Para ilustrar la problemática mencionada se toman como referencia casos de Japón y Chile donde se pudo evidenciar la magnitud de los daños y la incomunicación que sufrieron ambos países por horas, lo que contribuyó a una mayor pérdida de vidas humanas (McClelland, 2011).

Parte del territorio sudamericano tiene una alta probabilidad de movimientos telúricos de gran escala, esto debido a su pertenencia al cinturón de fuego del pacífico, que es donde se concentra gran cantidad de estos fenómenos a lo largo de la historia en el mundo. Como lo muestra la tabla 1, únicamente la región que abarca desde el sur de Chile hasta México posee un número elevado de sismos desde 1970 a 2014 (López et al., 2016).

La tabla 1 muestra una serie de zonas abarcadas por la costa pacífica del continente tomadas desde el norte de Chile hasta México, siendo la zona 7 la que empieza por el primer país mencionado y la zona 10 el último. Esta subdivisión se realiza con el objetivo de tener una perspectiva más clara de la cantidad de terremotos sufridos a lo largo de esta distribución geográfica.

Tabla 1. *Distribución de terremotos desde el norte de Chile hasta México.*

	1970-1974	1975-1979	1980-1984	1985-1989	1990-1994	1995-1999	2000-2004	2005-2009	2010-2014
Zona 7	6	6	5	14	4	10	13	12	30
Zona 8	6	1	2	6	5	5	4	13	10
Zona 9	2	6	4	4	11	12	18	17	7
Zona 10	2	3	4	4	10	20	12	11	11

Fuente. López et al., 2016

Según los datos mostrados, se puede deducir que los terremotos pueden ser bastante desastrosos en cualquier circunstancia, sin embargo, no es el único desastre que el planeta ha sufrido a lo largo de los últimos años. En la tabla 2 se muestran los datos de todos los desastres que se han sufrido en un periodo de 9 años, demostrando la gran cantidad de peligros inesperados que pueden ocurrir en cualquier momento.

Tabla 2. *Datos mundiales de desastres naturales y sus consecuencias en el periodo comprendido de 2003 a 2012.*

Fenómenos naturales	N.° de desastres	%	N.° de muertos	%	Daños en millones de US$	%
Total desastres naturales	3.885	100	1.066.119	100	1.545.540	100
Inundaciones	1.762	45.35	56.886	5.34	373.389	17.69
Tormentas de viento	990	25.48	175.241	16.44	659.846	42.69
Terremotos, tsunamis	282	7.26	678.818	63.67	487.061	31.51
Temperaturas extremas	273	7.03	144.714	13.57	43.245	2.80
Sequias, inseguridad alimentaria	234	6.02	424	0.04	51.372	3.32
Movimiento en masas húmedas	183	4.71	8.956	0.84	1.689	0.11
Incendios de matorral	98	2.53	717	0.07	28.767	1.86
Erupciones volcánicas	56	1.44	363	0.03	171	0.01

Fuente. (Capacci & Mangano, 2015)

Tabla 2. *(Continuación…).*

Fenómenos naturales	N.° de desastres	%	N.° de muertos	%	Daños en millones de US$	%
Movimiento en masas secas	7	0.18	n.d		n.d	

Es por todo esto que se llega a la necesidad de realizar un monitoreo de actividad humana, con el fin de llegar a posibles sobrevivientes de todos estos hechos, y así disminuir la tasa de mortalidad.

Para realizar el monitoreo, existen ya varios procesos que se han implementado a la medicina recientemente, para el monitoreo de actividad humana con el uso de sensores. Regularmente se monitorean los ejercicios que hace un paciente sin necesidad de acercarse a un centro médico, entre otros (Aguado, 2016).

Debido a que los medios de telecomunicaciones, tales como las redes móviles, solo tienen como soporte su infraestructura, y teniendo en cuenta que se han realizado investigaciones y aplicaciones acerca del reconocimiento de actividad humana con sensores, éstos se pueden complementar con fines más específicos, más aún viendo la cantidad elevada de desastres naturales con la que el territorio colombiano y los allegados cuentan. Esto muestra la probabilidad y el riesgo al que nos exponemos si continuamos con dichas restricciones.

Debido a la problemática expuesta anteriormente, en este trabajo de grado se propone el entrenamiento de varios modelos de aprendizaje automático para predecir si un teléfono móvil está siendo utilizado por un ser humano en determinado momento, haciendo la predicción a partir de las lecturas de los sensores del dispositivo. Cada modelo es evaluado con el fin de estimar cuál ofrece la predicción más confiable.

Como trabajo posterior, y por fuera de este trabajo de grado, se tiene previsto acoplar el mejor modelo a la aplicación Conecta2, adscrita al trabajo de

investigación, de tal forma que la app pueda predecir si un ser humano se encuentra utilizando un dispositivo inteligente, o, por el contrario, NO está utilizando este dispositivo, dando prioridad a la comunicación con dispositivos que estén en el primer caso ya que aquellos son los que tienen la mayor probabilidad de estar en manos de sobrevivientes de la catástrofe.

1.2. FORMULACIÓN DEL PROBLEMA

Actualmente en la literatura científica se pueden encontrar estudios sobre reconocimiento automático de actividad humana en dispositivos móviles, no se cuenta con modelos de predicción encaminados a la localización de posibles víctimas en desastres naturales con base en los sensores del teléfono inteligente. Esta carencia se da de forma particular en el marco del proyecto de investigación "Sistema de comunicación para sobrevivientes de un desastre basado en una red ad hoc de teléfonos inteligentes", en el cual no se tiene una aplicación móvil que permita hacer esta predicción en términos de los sensores que se tienen disponibles en la mayoría de los dispositivos.

Con este proyecto se buscó resolver la siguiente pregunta de investigación: ¿Cuál será el mejor algoritmo de clasificación para identificar si un teléfono inteligente está siendo usado por un ser humano, entre máquinas de soporte de vectores, bosques aleatorios y redes neuronales?

1.3. JUSTIFICACIÓN

Los desastres naturales son eventos que muy difícilmente se pueden predecir. Es cierto que hay países que son más propensos a ellos por su ubicación geográfica; por ejemplo, Estados Unidos con huracanes y Chile con terremotos por solo mencionar algunos. Esto hace que en cualquier momento de la vida cotidiana se pueda sufrir las consecuencias que dichos eventos conllevan.

Es por esto que se desarrolló este proyecto, para que independientemente del momento o lugar en el que una persona se encuentre, teniendo en consideración que es muy probable que la duración de la batería del dispositivo disminuya drásticamente, sea más fácil el acceso a rescates y el aumento de vidas salvadas en estas condiciones.

El proyecto se podrá ver implementado en ocasiones de desastres naturales, ahorrando costes de búsqueda y rescate en sitios donde posiblemente no se encuentren sobrevivientes, dando prioridad a aquellos que se sabe que en realidad podrían estar en situación de riesgo.

El factor benéfico en cuanto a la sociedad será bastante amplio, ya que como se mencionó anteriormente, nadie está excluido en caso de un desastre natural; de esta manera toda la población tendría un beneficio potencial con este proyecto.

En vista al desarrollo del proyecto, ya existen algoritmos para la detección de actividad humana; sin embargo, son de alcance muy global. Al especificar el uso del algoritmo que se realizó para detectar el uso de un teléfono inteligente, ayudará a la detección de posibles sobrevivientes bajo escombros en algún desastre, lo que sería imposible para una persona de manera manual o intuitiva.

Este proyecto contribuye ampliamente al desarrollo profesional del autor, ya que se está trabajando con técnicas de aprendizaje automático, que actualmente se están implementando en la mayoría de las actividades en las que nos vemos inmersos día a día. De igual manera el enriquecimiento de conocimientos en lenguajes de programación que hasta el momento eran muy poco conocidas o utilizadas, esto permite tener más cabida dentro del mercado profesional y brindar mejores soluciones a futuros proyectos.

1.4. OBJETIVOS

1.4.1. Objetivo general

Evaluar varios modelos de aprendizaje automático para identificar en algún momento cualquier teléfono inteligente con sistema operativo Android está siendo utilizado por un ser humano, a partir de las lecturas en los sensores del dispositivo, utilizando diversas parametrizaciones y técnicas de aprendizaje (redes neuronales, máquinas de vectores de soporte y bosques aleatorios) para identificar el más confiable en términos de las métricas de desempeño más comúnmente utilizadas en la literatura científica, incluyendo la curva ROC y el puntaje F1.

1.4.2. Objetivos específicos

1. Pre procesar el conjunto de datos base provisto por el proyecto de investigación, compuesto de lecturas de sensores en teléfonos inteligentes, con el fin de tener información adecuada para los modelos de aprendizaje, por medio de códigos en Python.

2. Entrenar los modelos de aprendizaje automático con cada técnica de clasificación, para realizar el análisis de los resultados a partir del conjunto de datos base pre procesados, con los códigos realizados en Python.

3. Identificar el modelo de aprendizaje con el mejor desempeño predictivo con base en los ya entrenados con los datos pre procesados, a partir de varias

métricas, incluyendo la curva ROC y el F1, teniendo como herramienta un código realizado en Python.

1.5. ALCANCE Y LIMITACIONES DEL PROYECTO

1.5.1 Alcance

- Los modelos se implementaron bajo el lenguaje de programación Python.
- Se utilizaron tres diferentes técnicas de aprendizaje las cuales serán máquinas de vectores de soporte, bosques aleatorios y redes neuronales.
- Para evaluar los modelos se usaron las métricas *Accuracy,* Curva ROC*, F1,* entre otras, buscando identificar el modelo que ofrezca las predicciones más confiables.

1.5.2 Limitaciones

- No se realiza soporte y/o mantenimiento sobre el código entregado.
- En caso de que la capacidad de un servidor local no soporte los procesos por el tamaño del *dataset* no se usan servidores o herramientas externas.
- El proyecto se limitó a realizar únicamente el sistema de predicción, el acople a la aplicación principal se podrá ver estipulado en otro trabajo de grado distinto.

2. MARCO DE REFERENCIA

2.1. MARCO TEÓRICO

El soporte teórico brindado a continuación dará al lector una comprensión más clara de lo que trata el trabajo, dando conceptos completos acerca de diferentes temáticas que abarcan la problemática y la solución con ejemplos para mayor facilidad de entendimiento.

2.1.1. Aprendizaje Automático

El cerebro humano es un ámbito de investigación importante. Desde hace mucho tiempo se ha intentado descubrir como es el funcionamiento interno del comportamiento desde el cerebro, aun, ahora con tanta tecnología no se ha logrado saber del todo como es que lo hace; sin embargo, se ha pensado la posibilidad que una máquina simule este comportamiento, con el fin de hacer labores tediosas del día a día como labores hogareñas, hasta un asistente médico (Briega, n.d.).

El termino de aprendizaje de máquina se refiere a sistemas complejos que se encargan de reconocer patrones dentro de una cantidad amplia de datos; esto con el fin de reconocer comportamientos futuros y mejorar con el tiempo, sin necesidad de ninguna intervención (CleverData, 2019).

Dentro del aprendizaje automático se encuentran tres tipos de aplicación:

- **Aprendizaje Supervisado:** es aquel que, dado una serie de datos acompañados con etiquetas (datos de entrenamiento), el sistema detecta patrones, para que posteriormente se puedan introducir datos sin dichas etiquetas y pueda identificar o predecir diferentes datos (datos de prueba), sin necesidad de información adicional (Redacción APD, 2019).

  Un ejemplo de aprendizaje supervisado podría ser la construcción de un modelo donde se busque predecir si una persona sufrirá de alguna enfermedad teniendo en cuenta sus hábitos, esto con una base de datos

previa de pacientes que han sufrido enfermedades que son consecuencia de malos hábitos.

- **Aprendizaje no supervisado:** como su nombre lo sugiere, es lo completamente opuesto al anterior. Su funcionamiento se basa en la abstracción y comprensión de patrones con información sin etiquetas.

  Todo este procesamiento se conoce como *clustering*, este aprendizaje se asimila más al funcionamiento de análisis de información en los humanos (Redacción APD, 2019).

  El ejemplo más claro y común de este tipo de aprendizaje es la segmentación de clientes en un almacén de cadena, según su rango de edad o sus hábitos de compras, aquí no se tiene en cuenta ningún tipo de información previa registrada.

- **Aprendizaje por refuerzo:** es más enfocado a cómo se maneja el aprendizaje de carros autónomos y *bots*. Con las decisiones por experiencia se da una clase de incentivo o castigo, es de este modo que aquí no es necesario brindarle una información específica. El sistema va aprendiendo según las decisiones o caminos que tome (Redacción APD, 2019).

  Esto se ve más en la naturaleza que en la tecnología; por ejemplo, cuando una persona estudia lo regular que busca es tener una buena nota, al obtenerla se esforzará por continuar con ese mismo rendimiento; cuando su nota no fue muy buena intenta cambiar los patrones de estudio que lleva, con el fin de mejorar su rendimiento.

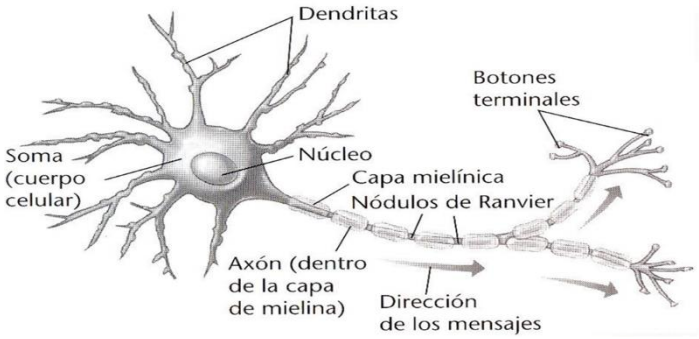
2.1.2 Redes Neuronales Artificiales

Una neurona humana suele ser representada como en la Figura 1, la cual es únicamente un modelo escalar. Si se analiza una sola neurona es poco eficiente comparándola con otros procesos existentes; sin embargo, el cerebro humano corrige estos errores generando cantidades enormes de conexiones entre gran

cantidad de igual manera de neuronas. Esto hace que se conforme una red compleja (Caicedo & López, 2010).

Su comportamiento para procesar toda la información comienza en el soma de la neurona, donde envía esta al axón. De ahí pasa a las dendritas de la siguiente neurona siguiendo así un proceso continuo (Caicedo & López, 2010).

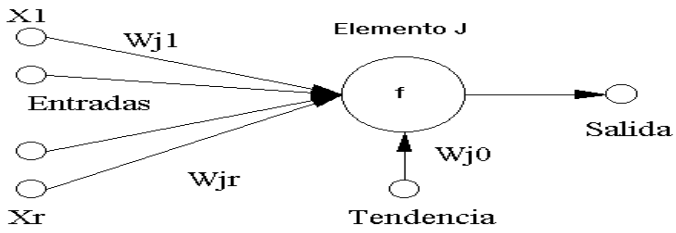
Ilustración 1. *Componentes de la neurona*



Fuente: *Las Neuronas*, n.d.

En el caso de las redes neuronales artificiales se intenta simular el comportamiento anteriormente descrito, siendo las entradas de información multiplicadas por el peso sináptico de cada neurona, las que simulan lo equivalente a un impulso nervioso en la neurona biológica (Ver figura 2). Este se procesa dentro de la neurona con alguna función de activación (Oliveira García-Ollala, 2019).

Ilustración 2. *Modelo tradicional de red neuronal monocapa.*



ELEMENTO SIMPLE DE PROCESADO

Fuente: *Fundamentos Básicos*, n.d.

Las funciones de activación son procesos que se realizan dentro de una neurona que determina su salida con base en la entrada que reciba. Para esto existen varios

tipos de funciones de activación, cada uno calcula la salida de diferente manera (Calvo, 2017). A continuación, se mencionan estos tipos de activación:

- **Sigmoidal:** se encarga de realizar un tipo de normalización de valores, estableciéndolos entre 0 y 1. Es más característico en la neurona de salida. Su función es la siguiente:

$$f(x) = \frac{1}{1 - e^{-x}}$$

- **Tangente hiperbólica:** es similar a la función sigmoidal, pero en este caso establece los valores entre -1 y 1. Se utiliza más en la toma de decisión entre uno y otro. Su función es la siguiente:

$$f(x) = \frac{2}{1 + e^{-2x}} - 1$$

- ***ReLu*:** su función es depurar los datos negativos, dejando los valores positivos sin modificación. Es más usado en el procesamiento de imágenes. Su función es la siguiente:

$$f(x) = \max(0, \mathrm{x}) = \begin{cases} 0 \; for \;\; x < 0 \\ x \; for \;\; x \geq 0 \end{cases}$$

- ***Leaky ReLu:*** similar a la función *ReLu* con la diferencia que toma los valores negativos y los multiplica por un coeficiente en vez de eliminarlos. Los datos positivos no sufren cambio. De igual manera procesa de buena manera las imágenes. Su función es la siguiente:

$$f(x) = \begin{cases} 0 \quad\quad for \;\; x < 0 \\ a \cdot x \; for \;\; x \geq 0 \end{cases}$$

- ***Softmax:*** en esta función lo que se transforma son las salidas, dándolas en medida probabilística, haciendo que la suma de todas las salidas tenga que

dar un total de 1. Se usa para normalizar datos cuando se trata de múltiples clases, y se desempeña bien en las últimas capas. Su función es la siguiente:

$$f(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

Un ejemplo de la aplicación de redes neuronales es la detección de rostros, sea en cámaras de vigilancia o en imágenes, como es el caso de Facebook, que en su red social realiza un reconocimiento facial, con el fin de realizar etiquetado automático.

En redes neuronales existen muchos tipos de redes, pero las más conocidas son:

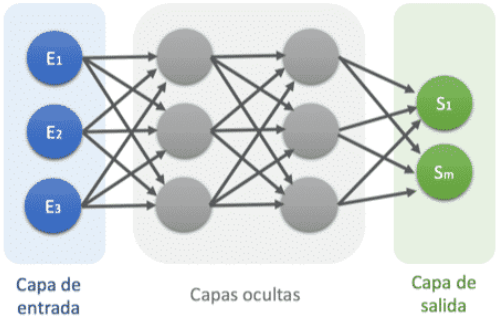
- **Red neuronal monocapa:** es la representación más sencilla, donde una o muchas neuronas reciben las entradas del sistema, datos que posteriormente envía a una única neurona donde se hacen los procesos con las funciones de activación, tal como lo muestra la figura 2.

  Para esta red los procesos de entrenamiento suelen ser más sencillos, pero a su vez son muy limitados.

- **Red neuronal multicapa:** esta red es más compleja que la anterior, ya que se compone de varias capas intermedias llamadas capas ocultas (Ver figura 3), las cuales corrigen errores que podía tener el contar con una única capa, permitiendo realizar procesos más complejos de aprendizaje.

  Así mismo, los procesos de entrenamiento suelen ser más complejos.

Ilustración 3. *Red neuronal multicapa*



Fuente: *Clasificación de Redes Neuronales Artificiales - Diego Calvo*, n.d.

2.1.3 Máquinas de Vectores de Soporte (SVM)

Las máquinas de vectores de soporte se usan como herramienta de clasificación dentro del aprendizaje automático. Su función es mapear los datos de entrada para generar una frontera entre ellos. El ideal es que esta frontera esté lo más alejado de los datos para que sea más preciso.

Una de las aplicaciones más reconocidas de las SVM es en el proceso OCR (*Optical Character Recognition),* es decir el reconocimiento de imágenes, ya que estas máquinas de vectores tienen un gran rendimiento procesándolas (Scientia Et Technica, 2005).

Las máquinas se utilizan generalmente para solucionar problemas de regresión y clasificación. Su mayor desventaja es la complejidad de los algoritmos que se aplican.

Estas máquinas manejan dos casos específicos para la clasificación:

- **Casos linealmente separables:** en estos casos se definen hiperplanos para cada tipo de datos. Los datos se separan y se les asignan las etiquetas 1 y -1, de tal manera que al clasificarlos se les pueda separar dentro del hiperplano asignado para cada una de las etiquetas.

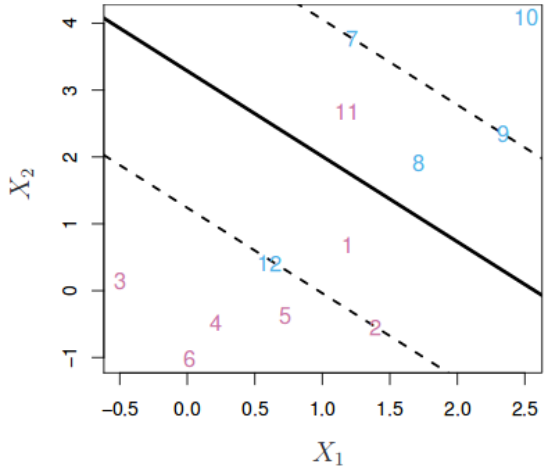
  El problema con estos casos es que hay una cantidad infinita de posibilidades para la distribución de los datos, así que hay que tener otro proceso adicional para determinar cuál es el más adecuado.

- **Casos NO linealmente separables:** estos casos son más comunes en la vida real, dado que no todos los datos son realmente separables fácilmente por una función lineal. Esto lleva a utilizar otro tipo de clasificador para solucionar estos problemas, llamados *Soft Margin classifiers*.

Estos clasificadores permiten realizar el mismo proceso de generar la limitante que realizan los casos anteriores, con la diferencia que no son tan estrictos. Permiten que unos pocos datos salgan del hiperplano, estos son considerados como mal clasificados, pero esto permite que se adapte más fácilmente a los cambios posibles dentro de la información, cosa que los clasificadores lineales no permiten (Rodrigo, 2017).

En la figura 4 se demuestra con un ejemplo como quedaría un proceso de clasificación con estas características.

Ilustración 4. *Clasificación en un caso NO linealmente separable.*



Fuente: *Máquinas de Vector Soporte (Support Vector Machines, SVMs)*, n.d.

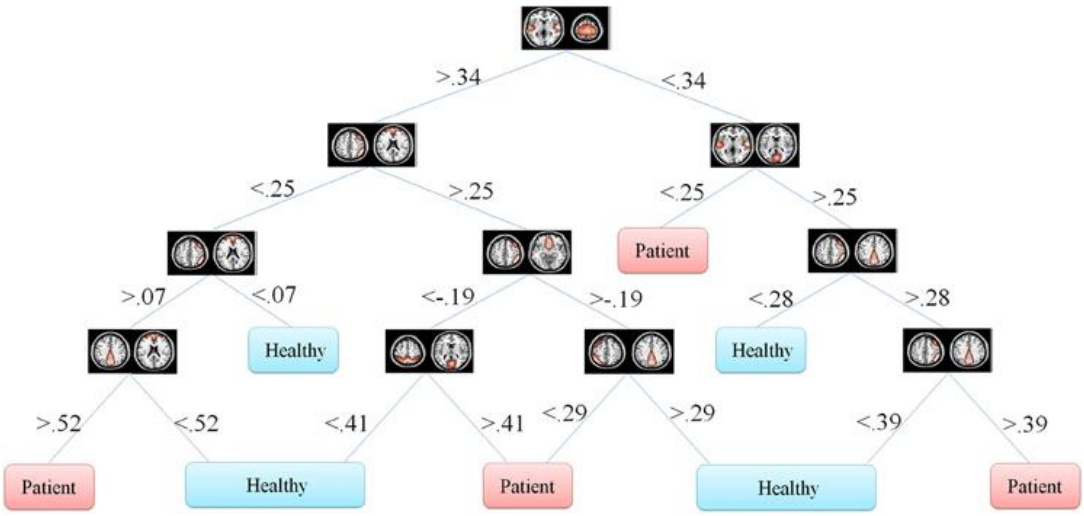
2.1.4 Bosques Aleatorios

Para lograr entender el funcionamiento de los bosques aleatorios primero hay que entender el concepto de árboles de clasificación, los cuales fueron creados con base en la jerarquía de un árbol, donde desde su raíz se puede ir navegando hasta sus hojas; este método se ha utilizado en varios campos de estudio, desde la economía hasta la inteligencia artificial.

En el caso del aprendizaje automático, se determina la elección del mejor nodo por medio de la ponderación de los caminos que se recorren, lo que se puede evidenciar en la figura 5. Este método de clasificación supervisada suele ser efectivo; sin

embargo, suele tener limitaciones que los bosques aleatorios resuelven (Medina-Merino & Ñique-Chacón, 2017).

Ilustración 5. *Ejemplo de árbol de decisión.*



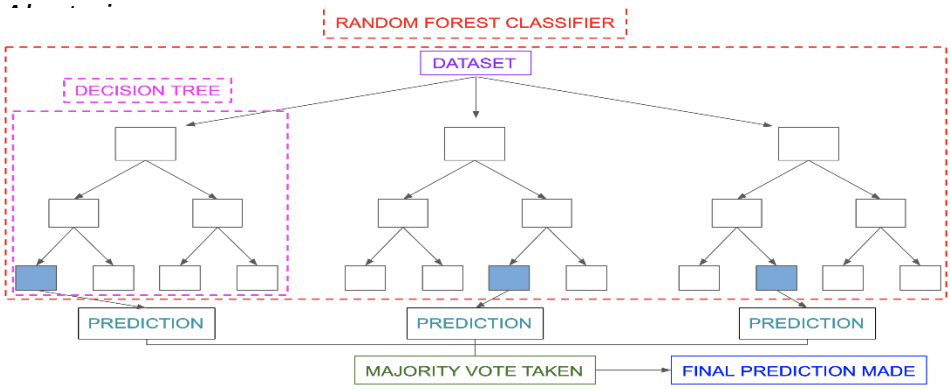
Fuente: (Caparrini, 2019)

Los bosques aleatorios son una combinación de los árboles de decisión, donde como su nombre lo indica, aleatoriamente se escogen datos del *dataset* seleccionado, con el fin de realizar el mismo proceso con un objetivo en común.

Se toman en cuenta distintos arboles de decisión, esto para tener una mayor efectividad, puesto que los errores que presenta uno de estos árboles no será el mismo que otro, debido a que no se entrenan con los mismos datos.

Es así, que los errores de uno se verán corregidos por el desempeño del otro; de esta manera se puede minimizar el error, y así obtener mejores resultados al final del procesamiento de los datos; esto se ve más claramente en la figura 6 (Medina-Merino & Ñique-Chacón, 2017)

Ilustración 6. *Diferencia entre Árbol de Decisión y Bosque*



Fuente: (Kashyap, n.d.)

2.1.5 Redes Ad Hoc

Las redes Ad Hoc son conexiones que realizan dispositivos electrónicos para realizar cierto tipo de actividades como, por ejemplo: comunicarse, auto configurarse entre otras, esto sin necesidad de una infraestructura de red previa.

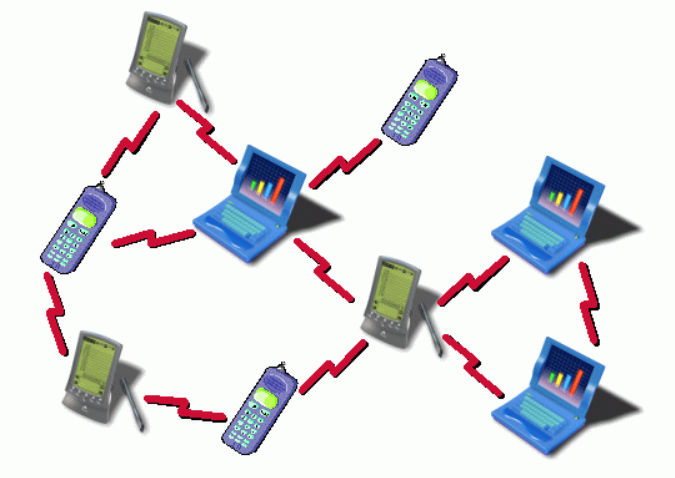
Algunas de las características de estas redes son que, a pesar de ser más comúnmente vistas en dispositivos móviles como tabletas o celulares, los dispositivos estáticos como computadores de mesa también poseen de estas redes. Sin embargo, el uso se ve más claramente en los primeros casos.

De esta manera al tener un constante movimiento puede que los nodos que se forman entre dos dispositivos pueden desaparecer y conectarse a su vez con otro nodo cercano. Esto hace que sea bastante variable, ya que al desconectarse un nodo de una red podría modificar el camino de acceso y contacto entre los demás.

Aun viendo todos estos beneficios, también tienen sus limitantes. Por ejemplo, los teléfonos celulares tienen un nivel de batería que regularmente disminuye entre más procesos realice. Esto hace que el alcance de la red se vea disminuido, aunque tienen una ventaja y es que los nodos actúan como repetidores, gracias a esto, como lo mencionado anteriormente genera un camino de acceso con el fin de tener comunicación sin verse afectado por este poco alcance (ver figura 7).

Estos se ven diferenciados de las redes tradicionales de internet debido a que estos últimos utilizan regularmente una arquitectura cliente-servidor, cosa que en este caso no aplica ya que ningún nodo implicado tendría el rol de servidor (Buran, 2004).

Ilustración 7. *Comportamiento de las redes Ad Hoc*



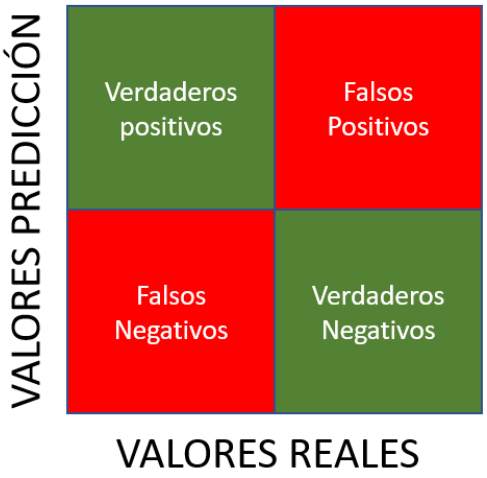
Fuente: *Orígenes de Las Redes Mesh I: Las Primeras Redes Ad-Hoc* | *Sevilla Mesh*, n.d.

2.1.6 Matriz de confusión

La matriz de confusión es una herramienta ampliamente usada dentro del campo de la inteligencia artificial, más específicamente en aprendizaje automático, y ayuda a determinar el desempeño de un algoritmo de clasificación.

Tiene cuatro campos: ‘Verdaderos positivos’, ‘Falsos positivos’, ‘Falsos negativos’, y ‘Verdaderos negativos’, como se evidencia en la ilustración 8 (InteractiveChaos, n.d.).

Ilustración 8. *Matriz de confusión*



Fuente: Barrios, 2019**.**

Los términos positivos y negativos se pueden entender como las variables que se van a predecir, de este modo si en el caso del trabajo de investigación se tiene que positivo puede referirse a cuando se tiene en uso el dispositivo inteligente, mientras que el negativo es cuando un dispositivo inteligente no se encuentra en uso por un individuo; de esta manera los 'Verdadero' sean positivos o negativos son aquellos que se predicen de manera correcta y los 'Falsos' sean positivos o negativos son aquellos que se predicen de manera incorrecta.

2.1.7 Dataset

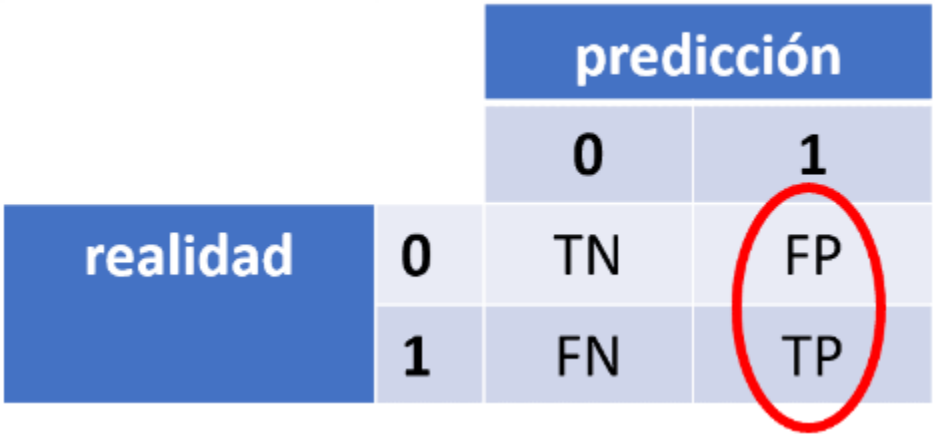
Un *Dataset* es un conjunto de datos regularmente tabulados, en los cuales se tiene una columna donde se describe la variable de la que se está hablando, y el resto de las columnas son los datos ya organizados. Estos datos pueden ser por un solo individuo-producto o por el contrario se pueden ver almacenados varios, dependiendo la cantidad de filas que este tenga (Balagueró, 2018).

2.1.8 Accuracy

Cuando se habla de *Accuracy* nos referimos al porcentaje de elementos correctamente clasificados dentro de un modelo aplicado. Se considera la forma más directa de evaluar la calidad del proceso de clasificación.

Se cuantifica con valores de 0 a 1 siendo el más cercano a 1 el mejor resultado, y el más cercano a 0 el menos óptimo. Sin embargo, no siempre es la mejor opción para implementar. Aunque regularmente todos los datos se encuentran balanceados, hay algunos casos en los que no, por lo que esta métrica seria alta pero no necesariamente nos demostraría que el modelo se encuentra realizando un buen entrenamiento (Sitiobigdata, 2019). La figura 9 permite observar la demostración del *Accuracy* en la matriz de confusión.

Ilustración 9. *Demostración gráfica del Accuracy en la matriz de confusión*



Fuente: Heras, n.d.

2.1.9 F1 Score

La métrica F1 realiza una combinación entre dos métricas de medición de rendimiento, *Recall* y Precisión, con el fin de tener un valor unificado entre la precisión y la exhaustividad (cantidad que el modelo puede identificar). No siempre es necesario utilizar esta opción, ya que no en todos los casos es lo mejor combinar ambas mediciones. Esto se debe observar de manera detallada dependiendo el *dataset* que se vaya a manipular (Heras, n.d.). La siguiente es la ecuación:

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

2.1.10 Recall

Esta métrica se refiere a la cantidad de los datos positivos que el modelo en cuestión es capaz de detectar sobre el total de los datos, esto puede dar una idea de que tan efectivo es nuestro modelo; se calcula teniendo en cuenta la matriz de confusión de la siguiente manera (Martínez, 2020).

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

2.1.11 Precision

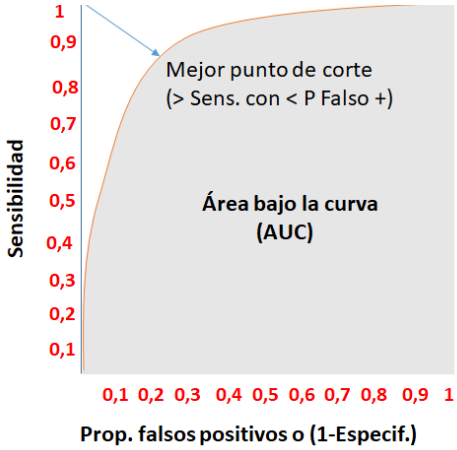
Nos ayuda a medir la calidad del modelo que estemos evaluando frente a unos datos, esto, dándonos respuesta a la cantidad porcentual de datos positivos que existen dentro de la información suministrada; para esto tenemos que tener en cuenta los datos de la matriz de confusión y se evalúan de la siguiente manera (Martínez, 2020).

$$precision = \frac{TP}{TP + FP}$$

2.1.12 Curva ROC

La curva ROC se usa para medir la calidad de la salida del modelo, ya que se basa en la tasa de falsos positivos y verdaderos positivos. Con esta curva podemos mantener una tasa de falsos positivos en 0 o muy cercano a este valor, mientras que los verdaderos positivos tienen que aproximarse a 1. De esta manera, se puede asegurar que el modelo se encuentra realizando un buen procesamiento de los datos (Abad, n.d.). La figura 10 permite visualizar una gráfica de ejemplo de la curva ROC.

Ilustración 10. *Ejemplo de la curva ROC*



Fuente: Oberto, n.d.

2.1.13 Principal Component Analysis(PCA)

Es un método estadístico que permite redimensionar un espacio muestral, con el fin de tener una cantidad de parámetros menor para optimizar el rendimiento de los modelos de aprendizaje, esto sin alterar de manera radical la información que es suministrada; es decir se realiza un análisis de impacto de los datos para así conservar la más importante, a esta información que se conserva se le llaman componentes principales (Amat, 2017).

Para la reducción de las dimensiones se pueden seguir 3 métodos: el primer método es escoger arbitrariamente las dimensiones que más relevancia nos parezca, la que puede ser una de las más inseguras; la segunda opción es por medio del cálculo de la proporción de variación explicada (Método estadístico que tiene en cuenta la dispersión de la información), característica por característica hasta llegar a un umbral definido; regularmente se puede llegar a un 85% o más de la variabilidad. Por último en el método 3 se realiza una gráfica donde visualmente se puede detectar la variación y se puede establecer un valor cercano (Na8, 2018).

2.1.14 Recursive Feature Elimination (RFE)

Es un método que por medio de la estadística realiza la supresión de características que puedan ser irrelevantes para un modelo de aprendizaje automático; este modelo, a diferencia del PCA se caracteriza por intentar eliminar dependencias y colinealidad que pueda existir en la información.

Este proceso le asigna un valor o coeficiente a cada una de las columnas por importancia, una vez establecida la cantidad de columnas necesarias se descartan las de menor coeficiente con el fin de reajustar la cantidad de datos (Yellowbrick, n.d.).

2.1.15 Scikit-learn

Es una librería que se usa en aprendizaje automático que se encuentra disponible para el lenguaje de programación Python, esta contiene las técnicas de clasificación, reducción de dimensiones y otras que no se usarán en este trabajo; gracias a esto, es una herramienta fundamental para realizar análisis de datos por su facilidad de uso y su acople frente a otras librerías del mismo de lenguaje de programación.

2.1.16 GridSearchCV

Es una herramienta de selección de variables, la cual viene incluida en la biblioteca *Scikit-learn*, la cual realiza una determinación de valores óptimos entre parámetros propios de cada modelo de aprendizaje automático; de esta manera se logra verificar qué combinación es adecuada para obtener un mejor rendimiento en un menor tiempo que si se hiciera manualmente.

Para la utilización de esta herramienta se define que parámetros se van a tener en cuenta para cada modelo y los posibles valores que pueden tomar dentro del entrenamiento. Teniendo esta definición, lo que hace la herramienta es realizar el entrenamiento con cada uno de ellos y evaluar sus resultados, para así escoger el más adecuado, y con la combinación adecuada realizar la muestra de resultados final (Medium, 2019).

Ilustración 11. *Ejemplo de elección de parámetro por GridSearchCV*

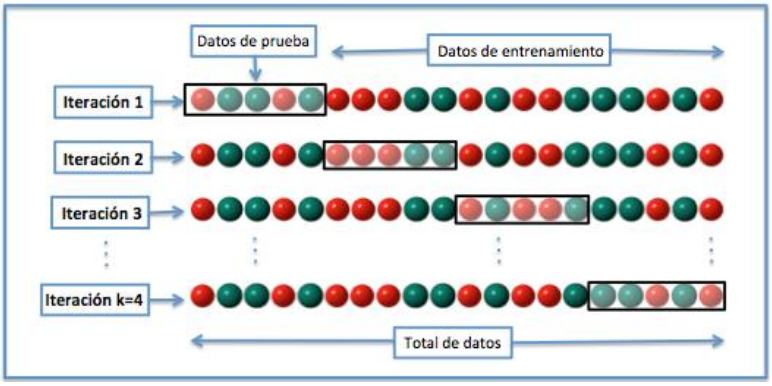
C	Alpha: 0.1	0.2	0.3	0.4
0.5	0.701	0.703	0.697	0.696
0.4	0.699	0.702	0.698	0.702
0.3	0.721	0.726	0.713	0.703
0.2	0.706	0.705	0.704	0.701
0.1	0.698	0.692	0.688	0.675

Fuente: DataTask, n.d.

2.1.17 Validación Cruzada

Es un modelo iterativo de entrenamiento que separa k grupos de un tamaño similar, los cuales se usan para dos procesos, k-1 grupos se usan para el entrenamiento y el restante para la validación, esto se realiza varias veces hasta que cada grupo se use para validación, no todas las estimaciones de error son iguales en cada iteración así que como estimación final se utiliza el promedio de estos (Amat Rodrigo, 2020).

Ilustración 12. *Funcionamiento de la validación cruzada*



Fuente: Wikipedia, n.d.

2.2. ANTECEDENTES

En el campo de la inteligencia artificial se han creado varios sistemas que con ayuda de los sensores de diversos dispositivos logran predecir la actividad humana. Sin embargo, algunos de estos eran muy invasivos, lo que hacía que un movimiento no fuera completamente natural y entorpeciera las lecturas. Se contempló el uso de los sensores de un teléfono inteligente, el cual es un objeto que tenemos a la mano en todo momento.

En 2013 se implementó la recolección de datos de sensores, con una cantidad de 30 voluntarios entre los 19 y 48 años, quienes tenían que portar un celular y realizar diferentes actividades. Finalmente se predijeron con un alto nivel de efectividad 6 actividades que se muestran en la tabla 3, generando un *dataset* de dominio público (Anguita et al., 2013).

Tabla 3. *Resultados de la predicción de actividad*

	WK	WU	WD	ST	SD	LD	**Recall**
Walking	**492**	1	3	0	0	0	99%
W. Upstairs	18	**451**	2	0	0	0	96%
W. Downstairs	4	6	**410**	0	0	0	98%
Sitting	0	2	0	**432**	57	0	88%
Standing	0	0	0	14	**518**	0	97%
Laying Down	0	0	0	0	0	**537**	100%
Precision	96%	98%	99%	97%	90%	100%	**96%**

Fuente: Anguita et al., 2013

Por otro lado, en el mismo año se realizó otra prueba, esta vez con la finalidad de predecir los accidentes de las personas mayores al momento de levantarse de la cama, de esta manera evitar posibles accidentes. Esta vez se utilizó un parche colocado en la ropa con sensores ubicados en él para captar su posición y ángulo. Se realizó en 14 personas voluntarias entre los 66 y 86 años, dichos experimentos se realizaron en cuartos distintos, permitiendo evaluar posibles variaciones por su locación.

En la tabla 4 se muestran los datos recolectados en ambos *datasets.* De igual manera se evaluaron las comodidades que tuvieron los voluntarios portando la prenda con el sensor, concluyendo con resultados aceptables, obteniendo resultados que en resumen muestran que el dispositivo no es invasivo (Shinmoto Torres et al., 2013).

Tabla 4. *Resultados obtenidos en los dos datasets dados.*

Method	Dataset	Accuracy	Sensitivity	Specificity
BEAS	RoomSet1	84.55±6.78	78.24±12.6	87.33±5.35
	RoomSet2	86.9±8.85	90.14±13.47	86.57±11.11
Baseline [16]	RoomSet1	68.28±5.9	14.45±15.4	91.14±5.85
	RoomSet2	69.6±8.36	19.02±19.72	93.36±4.8

Fuente: Shinmoto Torres et al., 2013

En otro caso, se realizó el estudio de actividad humana detectando la caminata por medio de teléfonos y relojes inteligentes. Debido a que estos últimos tuvieron una inserción más tardía en el mercado, existen menos modelos entrenados con este tipo de dispositivo.

Para ambos casos se usó como *dataset* las lecturas de los sensores de ambos dispositivos conocidos como giroscopio y acelerómetro. Se tomaron dos diferentes pruebas, una con 10 ejemplos y otra con 5, como se observa en las tablas 5 y 6,

donde se evidencia que el mejor sensor en ambos casos (Teléfono y reloj inteligentes) es el acelerómetro, brindando mejores datos de precisión.

Para esto se nota un cambio entre los dos muestreos, dando a entender que entre más sean los datos que se tengan, mayor será la precisión de entrenamiento (Weiss et al., 2019).

Tabla 5. *Resultados con 10 ejemplos*

Activity	Phone		Watch		Phone	Watch	Accels	Gyros	All
	accel	gyro	accel	gyro					
Walking	11.2	11.3	17.5	18.8	9.3	16.1	12.6	10.2	7.9
Jogging	11.5	13.2	18.1	19.3	10.3	15.1	11.3	13.8	9.8
Stairs	12.3	16.4	24.3	26.1	11.8	21.6	13.9	16.5	13.5
Sitting	13.6	26.3	21.8	33.4	12.8	22.3	10.7	27.2	13.0
Standing	14.7	26.0	22.6	33.3	15.6	23.0	11.9	27.9	15.4
Kicking	12.5	18.5	21.8	26.7	11.5	21.1	13.8	16.7	14.0
Dribbling	12.2	19.9	18.9	21.0	12.7	17.9	11.2	15.7	12.0
Catch	10.8	20.3	20.6	20.8	13.4	16.7	12.1	17.2	12.2
Typing	11.5	19.4	16.8	26.2	11.3	18.0	10.4	19.0	8.7
Writing	13.3	19.4	15.3	27.1	12.3	15.6	11.2	18.5	10.8
Clapping	11.3	20.5	15.8	20.8	11.7	19.2	9.7	14.6	10.6
Teeth	11.8	19.7	18.6	22.7	12.1	17.2	11.4	19.9	12.2
Folding	11.4	16.6	19.6	24.7	12.3	17.1	8.3	17.0	10.9
Pasta	12.4	23.0	18.4	28.8	14.4	20.4	12.3	22.6	10.9
Soup	9.6	22.4	17.6	24.6	10.1	17.5	8.6	21.7	9.8
Sandwich	11.4	22.6	24.1	30.2	10.4	22.1	10.1	23.6	12.3
Chips	12.3	23.3	19.2	29.5	11.7	20.3	11.3	20.4	10.2
Drinking	12.0	24.2	20.0	30.1	12.9	20.1	11.8	19.7	12.4
Ave	12.0	20.2	19.5	25.8	12.0	19.0	**11.3**	19.0	**11.5**

Fuente: Weiss et al., 2019

Tabla 3. *Resultados con 5 ejemplos*

Activity	Phone		Watch		Phone	Watch	Accels	Gyros	All
	accel	gyro	accel	gyro					
Walking	9.4	9.8	13.2	17.2	8.8	13.9	11.3	10.0	6.8
Jogging	7.8	10.8	16.2	15.2	9.7	12.7	9.0	11.2	8.3
Stairs	13.4	12.5	19.3	23.9	9.3	18.9	8.4	14.1	6.9
Sitting	10.4	23.7	14.5	32.1	8.8	17.0	10.0	21.1	10.2
Standing	12.1	22.1	16.7	31.6	10.9	15.2	10.0	21.5	7.7
Kicking	10.6	19.4	21.0	24.1	11.0	16.6	10.1	18.8	11.0
Dribbling	10.3	21.0	16.4	16.1	9.7	14.5	10.0	11.8	11.5
Catch	9.7	19.3	16.3	15.5	10.0	14.9	9.3	13.9	10.0
Typing	8.3	15.4	13.0	20.7	8.9	14.0	8.6	13.3	8.8
Writing	8.7	15.7	10.7	21.3	9.2	11.6	9.0	16.0	10.1
Clapping	9.4	13.4	12.9	17.2	10.1	13.2	8.1	14.8	8.5
Teeth	10.1	14.0	13.3	20.0	10.2	14.4	10.8	14.9	8.2
Folding	7.9	18.6	17.0	23.4	10.0	17.3	8.1	16.2	7.1
Pasta	8.0	23.7	14.3	26.6	8.9	18.5	9.0	19.6	5.4
Soup	7.3	19.2	17.0	22.3	6.1	13.3	7.8	17.5	8.0
Sandwich	9.9	17.9	17.5	25.7	11.4	17.7	8.2	16.2	9.3
Chips	9.9	21.5	14.7	25.9	10.3	18.1	8.5	17.2	8.0
Drinking	11.3	19.2	16.6	25.1	10.2	13.9	10.9	19.9	8.1
Ave	9.7	17.6	15.6	22.4	9.6	15.3	**9.3**	16.0	**9.3**

Fuente: Weiss et al., 2019

Para finalizar, el proyecto que se está trabajando en el marco de este documento, tiene como particularidad el hecho de realizar el estudio de todos los sensores del teléfono inteligente, con el cual se harán lecturas y se intentará predecir de la mejor manera posible la ubicación de una persona en riesgo potencial posterior a un desastre natural.

Lo anterior se realizará detectando si un ser humano se encuentra usando o no el dispositivo, caso que, como se vio en los casos anteriores se aproxima, pero no es concretamente con este fin, ya que se centran en actividades de la vida diaria sin enfocarse como tal en el uso específico del teléfono inteligente.

2.3. MARCO LEGAL

Con base en el marco legal colombiano, y con el fin de ceñirse al mismo, se da una descripción de las leyes que se tendrán en consideración a lo largo de la realización del proyecto, de igual manera las licencias que se utilizarán para el diseño y demás.

2.3.1 Leyes

- Ley 48 de 1975 (Ley de propiedad intelectual)

Por la cual se adhiere al país la convención universal de derechos de autor, la cual ratifica el derecho de los estados a respetar los derechos de autor de cualquier obra que se realiza dentro o fuera del propio país (Organización de las Naciones Unidas para la Educación la Ciencia y la Cultura, 1952; República, n.d.).

- Ley 1581 de 2012 (Habeas Data)

Esta ley se refiere al derecho que tienen las personas de controlar la información que empresas tienen sobre ellos, dándoles la potestad de poder exigir el acceso y la modificación de esta. De igual manera pueden restringir la libre distribución de esta, con el fin de evitar el uso de esta información con destinos diferentes a los brindados originalmente (Colombia, n.d.).

Este trabajo respetó lo estipulado por las leyes dispuestas anteriormente.

2.3.2 Licencias

- Licencia PSFL (*Python Software Foundation License*)

Es la licencia actual del lenguaje de programación Python. Es considerada de software libre, no es copyleft, lo que permite la modificación de su código fuente(Python Software Foundation, 2016).

- Licencia BSD 3 clausulas (*Berkeley Software Distribution*)

Esta versión, modificada de la licencia BSD original, permite la distribución sin límite alguno y sin restricción de propósito. Las únicas condiciones que impone es que se mantengan los avisos de derechos de autor y la renuncia de garantía. Aparte se prohíbe la utilización de nombres de contribuyentes con fines de aprobación de trabajos.

Este trabajo implementó las licencias mencionadas en este documento.

3. ASPECTOS METODOLÓGICOS

3.1 Hipótesis

Las lecturas de los sensores se pueden utilizar para predecir confiablemente si cualquier dispositivo está siendo utilizado por un ser humano, o no.

3.2 Método de recolección de datos

Bajo el proyecto de investigación se realizó una aplicación que se llama *SensorReader*, el cual por medio de lecturas de sensores que se encuentran inmersos en los dispositivos móviles, a partir de esto, teniendo en cuenta las actividades mencionadas en la ilustración 13, los 6 individuos que aportaron en esta recolección realizaron dichas acciones con el dispositivo y de ahí se guardaban los datos en un archivo con extensión CSV, la cual se almacenaba y renombraba en la nube de Google Drive para tener un soporte de esta.

Ilustración 13. *Actividades realizadas para la recolección de datos*

a. Ubicar el teléfono en una posición estática sobre alguna superficie en un espacio interior. Hacerlo por 4.5 minutos.
b. Ubicar el teléfono en una posición estática sobre alguna superficie en un espacio exterior. Hacerlo por 4.5 minutos.
c. Sostener el teléfono en un bolsillo o maletín mientras se está de pie o sentado. Hacerlo por 4.5 minutos.
d. Sostener el teléfono en un bolsillo o maletín mientras se camina. Hacerlo por 4.5 minutos.
e. Utilizar el teléfono mientras se está acostado en la posición más habitual o cómoda. Hacerlo por 3 minutos.
f. Utilizar el teléfono mientras se está sentado en la posición más habitual o cómoda. Hacerlo por 3 minutos.
g. Utilizar el teléfono mientras se está de pie en la posición más habitual o cómoda. Hacerlo por 3 minutos.
h. Utilizar el teléfono mientras se está caminando en la posición más habitual o cómoda. Hacerlo por 3 minutos.
i. Utilizar el teléfono mientras se suben escaleras en la posición más habitual o cómoda. Hacerlo por 3 minutos.
    - Se puede hacer pausadamente, con descansos intermedios de no más de 10 segundos.
j. Utilizar el teléfono mientras se bajan escaleras en la posición más habitual o cómoda. Hacerlo por 3 minutos.
    - Se puede hacer pausadamente, con descansos intermedios de no más de 10 segundos.

Fuente: Elaboración propia.

Como se evidencia en la imagen anterior, del literal a al literal d son las actividades que evidencian ausencia de actividad humana, mientras que del literal e al literal j se detecta presencia de actividad humana, es por esto que las primeras tienen un rango de tiempo más amplio, mientras que las últimas tienen un rango de tiempo más reducido, con el fin de tener una uniformidad en los datos recolectados y finalizar con un proceso de entrenamiento óptimo y sin mayores imprecisiones.

Se ve una diferencia de duración en las actividades en la ilustración 13, las cuales se realizan con el fin de tener una cantidad similar de datos tanto para los casos de ausencia de actividad humana como para los de presencia de actividad humana, de esta manera se evita cualquier sesgo en las métricas de desempeño al trabajar con datos balanceados.

Se tuvo en cuenta únicamente un total de 6 individuos debido a la pandemia por la cual se está pasando en este momento, ya que algunas de las actividades previamente mencionadas se requerían realizar al aire libre; por otro lado, la adición de más información por parte de más participantes hubiera implicado un requerimiento de hardware más potente, debido a que todos los procesos de los modelos frente a los datos se realizaron en una máquina personal, esto implicaría un coste de tiempo mucho mayor e ineficaz para la implementación.

3.3 Datos previos al pre procesamiento

Los datos antes de realizar el pre procesamiento eran, en cuanto a cantidad bastante amplios, es por esto que por medio de dos técnicas de selección de variables se comprimieron o eliminaron (según si fueran la técnica PCA o RFE), esto para disminuir las variables independientes y así facilitar el proceso de entrenamiento de los modelos de aprendizaje automático; de igual manera se tenía que realizar unos cálculos frente a los datos que se aclararán más adelante en el documento debido a que no brindaban una claridad concisa.

Los registros capturados poseían un total de 14 columnas distribuidos de la siguiente manera:

- Hora: se refiere a la fecha y hora de la recolección de los datos, los datos tienen una estructura DD/MM/AAAA HH: MM.

- Acelerómetro: se refiere a los registros capturados en el sensor del acelerómetro, cuenta con X, Y, Z, que se refieren a las 3 dimensiones; los datos recogidos por estos sensores son en formato decimal, tanto positivos como negativos.

- Giroscopio: se refiere a los registros capturados en el sensor del giroscopio, cuenta con X, Y, Z, que se refieren a las 3 dimensiones; los datos recogidos por estos sensores son en formato decimal, tanto positivos como negativos.

- Magnetómetro: se refiere a los registros capturados en el sensor del magnetómetro, cuenta con X, Y, Z, que se refieren a las 3 dimensiones; los datos recogidos por estos sensores son en formato decimal, tanto positivos como negativos.

- Proximidad: se refiere a los registros del sensor de proximidad del dispositivo móvil, los datos recogidos por estos sensores son en formato decimal, únicamente valores positivos.

- Latitud-longitud: esto se refiere a los registros recolectados del GPS del dispositivo móvil; los datos recogidos por estos sensores son en formato decimal, tanto positivos como negativos.

- Batería: es el porcentaje de batería que tiene el dispositivo al momento del registro de datos; los valores que aquí se encuentran están dados por un número porcentual.

El número de voluntarios seleccionados para hacer la toma de muestras fue de 6 individuos, los cuales realizaron las acciones mencionadas anteriormente, recolectando la información que se precisa para la realización del aprendizaje de los modelos.

En la ilustración 14 se puede observar cómo se almacenan las columnas en uno de los documentos CSV que se trabajaron.

Ilustración 14. *Ejemplo de los datos recolectados con SensorReader*

Hora	Acelerometr	Acelerometr	Acelerometr	Giroscopio X	Giroscopio Y	Giroscopio Z	Magnetome	Magnetome	Magnetome	Proximidad	Latitud	Longitud	Bateria
03/02/2021 10:35	-0.30765492	900.579	-4.570.531	0.017816897	0.22856541	0.009183341	10.199.997	-23.999.996	26.400.002	8.0	46.333.704	-7.406.442	67.0%
03/02/2021 10:35	-0.046686932	9.200.917	-56.000.376	0.017816897	0.22856541	0.009183341	10.899.994	-21.000.004	26.099.998	8.0	46.333.704	-7.406.442	67.0%
03/02/2021 10:35	-0.5781997	8.797.494	-58.705.826	0.017816897	0.22856541	0.009183341	10.899.994	-21.000.004	26.099.998	8.0	46.333.704	-7.406.442	67.0%
03/02/2021 10:35	0.046686932	8.647.857	-5.009.867	0.017816897	0.22856541	0.009183341	10.899.994	-21.000.004	26.099.998	8.0	46.333.704	-7.406.442	67.0%
03/02/2021 10:35	0.16639702	8.645.462	-44.699.745	0.017816897	0.22856541	0.009183341	10.899.994	-21.000.004	26.099.998	8.0	46.333.704	-7.406.442	67.0%
03/02/2021 10:35	0.31004912	8.664.616	-4.495.114	0.017816897	0.22856541	0.009183341	10.899.994	-21.000.004	26.099.998	8.0	46.333.704	-7.406.442	67.0%
03/02/2021 10:35	0.37588966	8.750.807	-431.076	0.017816897	0.22856541	0.009183341	10.899.994	-21.000.004	26.099.998	8.0	46.333.704	-7.406.442	67.0%
03/02/2021 10:35	-0.023942016	8.678.981	-4.201.824	-0.14956018	0.123496585	-0.103215866	10.899.994	-21.000.004	26.099.998	8.0	46.333.704	-7.406.442	67.0%
03/02/2021 10:35	-0.49559975	8.578.424	-4.578.911	-0.14956018	0.123496585	-0.103215866	10.899.994	-21.000.004	26.099.998	8.0	46.333.704	-7.406.442	67.0%
03/02/2021 10:35	-0.21787235	8.750.807	-4.654.328	-0.14956018	0.123496585	-0.103215866	11.300.003	-2.0	25.700.005	8.0	46.333.704	-7.406.442	67.0%
03/02/2021 10:35	-0.2801216	8.639.477	-47.141.833	-0.14956018	0.123496585	-0.103215866	11.300.003	-2.0	25.700.005	8.0	46.333.704	-7.406.442	67.0%

Fuente: Elaboración propia.

3.4 Datos posteriores al pre procesamiento

En el pre procesamiento realizado sobre los datos descritos anteriormente y que se explica en la sección 3.5.1, se realizó una disminución de columnas irrelevantes para el proceso de aprendizaje automático; de igual manera, se adicionaron columnas que corresponden a las agrupaciones y cálculos correspondientes al proceso; de esta manera quedan datos en archivo CSV listos para aplicarse a los modelos de aprendizaje.

Para efectos de este proyecto de investigación, los parámetros para los modelos se distribuyeron de la siguiente manera por cada uno:

SVM:

- **C:** es una penalización que se asigna por cada dato mal clasificado.
- **Gamma:** es la distancia que hay entre puntos, entre menor sea el valor más datos se agruparán.
- **Kernel:** es la función matemática que permite convertir problemas que no son lineales, a problemas lineales que ayudan a clasificar un modelo.

RNA:

- **Hidden_layer_sizes:** se refiere a la cantidad de capas ocultas que va a tener la red neuronal.
- **Activation:** esta es la función matemática de activación por la cual los datos se tratarán con el fin de realizar la predicción.
- **Solver:** es un modelo de optimización de los pesos que toma cada nodo en la red neuronal.

- **Alpha:** es una penalización que se asigna por cada dato mal clasificado.
- **Learning_rate:** este se refiere a la catalogación de la actualización de pesos.

Bosques Aleatorios:

- **N_estimators:** se refiere al número de árboles de decisión dentro del modelo.
- **Criterion:** es el criterio con el cual se divide cada nodo o rama.
- **Max_depth:** es la cantidad de nodos que tendrá cada árbol individual en cuanto a profundidad.
- **Min_samples_split:** es el número mínimo de datos para realizar la división de los mismos.

Los registros finales poseen un total de 55 columnas de las que se encuentran las siguientes:

- Max_(acc-mag-gyr): corresponde al cálculo del valor máximo de los datos agrupados, cuenta con X, Y, Z, que se refieren a las 3 dimensiones; los datos recogidos por estos sensores son en formato decimal, tanto positivos como negativos.

- Var_(acc-mag-gyr): corresponde al cálculo de la varianza de los datos agrupados, cuenta con X, Y, Z, que se refieren a las 3 dimensiones; los datos recogidos por estos sensores son en formato decimal, únicamente positivos.

- Standdev_(acc-mag-gyr): corresponde al cálculo de la desviación estándar de los datos agrupados, cuenta con X, Y, Z, que se refieren a las 3 dimensiones; los datos recogidos por estos sensores son en formato decimal, únicamente positivos.

- Min_(acc-mag-gyr): corresponde al cálculo del valor mínimo de los datos agrupados, cuenta con X, Y, Z, que se refieren a las 3 dimensiones; los datos recogidos por estos sensores son en formato decimal, tanto positivos como negativos.

- Mean_(acc-mag-gyr): corresponde al cálculo del promedio de los datos agrupados, cuenta con X, Y, Z, que se refieren a las 3 dimensiones; los datos recogidos por estos sensores son en formato decimal, tanto positivos como negativos.

- Absoldev_(acc-mag-gyr): corresponde al cálculo del promedio de la diferencia entre los datos agrupados, cuenta con X, Y, Z, que se refieren a las 3 dimensiones; los datos recogidos por estos sensores son en formato decimal, tanto positivos como negativos.

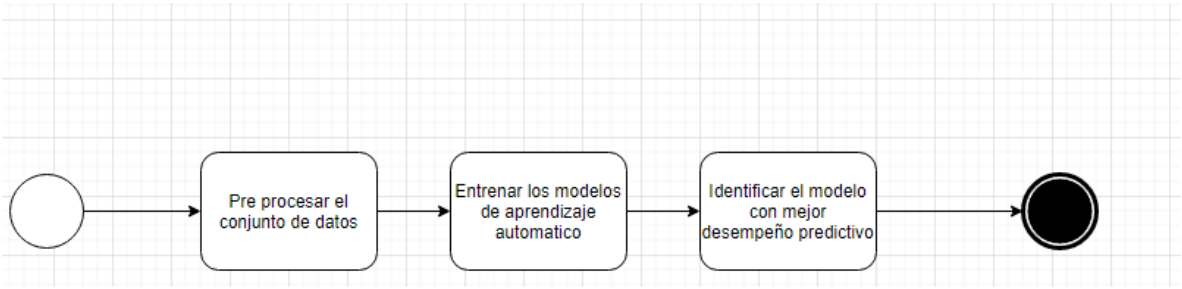
- Class: corresponde a la categorización de la actividad realizada en cada archivo, estos datos son de tipo entero y pueden tomar valores de 0 y 1.

Como es habitual en este tipo de proyectos, se define una variable dependiente la cual categoriza los datos de entrenamiento para el aprendizaje, y un conjunto de variables independientes que son las que definen posibles patrones para la predicción, para fines de este proyecto específico se utilizan los datos en la columna *Class* como variable independiente, y el restante de las columnas se utiliza como variables dependientes.

3.5 Diseño de investigación

Se trabajó bajo una metodología diseñada para este proyecto en específico, debido a que se deben cumplir con los objetivos anteriormente mencionados, tomando un único *dataset* recolectado con anterioridad por el trabajo de investigación, y dividiendo este en dos partes: una de ellas para entrenamiento y la otra para prueba de los modelos; se contempló con el fin de comprobación de los datos una validación cruzada de 5 y 10 iteraciones. En el modelo BPMN (Business Process Model and Notation) el cual es una notación estándar para modelado de procesos de negocio se ve evidenciado el paso a paso del proyecto. Ver figura No. 15.

Ilustración 15. *Modelo BPMN de las actividades en el proyecto.*



Fuente: Elaboración propia.

3.5.1 Pre procesamiento del conjunto de datos.

Respecto a los datos recolectados por el grupo de investigación, que se encuentran en una carpeta de drive, se realizó un script en Python que permite categorizar las carpetas con los archivos en formato csv (separados por comas), teniendo en cuenta que se poseen dos posibles predicciones; si el individuo se encuentra manipulando el dispositivo, o por el contrario no se encuentra en uso; así se subdividieron todos los archivos adicionando el atributo de clase que los caracteriza para el aprendizaje automático.

Realizado el proceso anterior, se generó una nueva carpeta que tiene los archivos ya procesados llamada "*Transformed_data*", con base en estos se realizó otro script en Python para la transformación de los datos.

La transformación de los datos se compone de una serie de cálculos que se hicieron para la comprensión de la información obtenida, los cuales se caracterizan por agruparlos en una ventana de tiempo, la cual se definió de 2.5 segundos debido a que es un tiempo habitualmente usado en la literatura académica investigada, por otro lado, también es necesario realizar el etiquetado de los datos, de este modo el modelo podrá detectar los casos en los cuales hay presencia de actividad humana, o por el contrario ausencia de actividad humana; los cálculos mencionados son los siguientes:

- **AVG:** se calcula obteniendo el promedio en cada eje de cada sensor en las ventanas de tiempo estipuladas.
- **Absoldev:** se calcula obteniendo el promedio entre la diferencia de los datos en cada eje de cada sensor y el promedio calculado.
- **Standdev:** se calcula obteniendo la desviación estándar en cada eje de cada sensor en las ventanas de tiempo estipuladas.
- **Var:** se calcula obteniendo la varianza en cada eje de cada sensor, en este caso se realiza una validación ya que este cálculo depende de dos valores o más, en caso de no tener dicha cantidad se estipula un valor de 0, de lo contrario se evalúa normalmente.
- **Min:** se calcula obteniendo el valor mínimo en cada eje de cada sensor en las ventanas de tiempo estipuladas.

- **Max:** se calcula obteniendo el valor máximo en cada eje de cada sensor en las ventanas de tiempo estipuladas.

3.5.2 Entrenamiento de los modelos de aprendizaje automático.

Como se tenía convenido anteriormente, se utilizaron los modelos de aprendizaje automático de SVM (‘*Support Vector Machine*’), Redes neuronales artificiales y Bosques aleatorios, los cuales fueron surtidos con el documento de extensión csv (archivo separado por comas), donde se encuentra la información agrupada de todos los sensores de todos los participantes.

El archivo donde se encuentra la información consolidada es bastante extenso en cuanto a cantidad de columnas, lo cual puede representar una dificultad para los modelos, es por esto que se definió un número de parámetros para trabajar, sin que la predicción se viera afectada; se acordaron 6 parámetros para trabajar debido a que se realizó el cálculo de la varianza y se evidenció que con esta medida se alcanza un aproximado de 90% de confiabilidad.

La información suministrada paso por una normalización estándar, la cual por cada variable independiente que se definió para cada modelo le asigna un valor de 0 y 1 a el promedio y la desviación estándar respectivamente, esto se realizó bajo la clase *StandardScaler* de la librería *scikit-learn.* Esta normalización se realiza con el fin de optimizar el proceso de aprendizaje de los modelos al darle datos con una varianza semejante, ya que si la varianza sobre un dato es mayor que otra el estimador puede no aprender de algunas características del sistema (Https://scikit-learn.org/, n.d.).

En cuanto a la selección de variables, en los tres modelos se contemplaron dos, PCA (‘*Principal Component Analysis*’), el cual es un cálculo estadístico que permite tener una reducción en las dimensiones de la información sin llegar a afectarla, y RFE (‘*Recursive Feature Elimination*’), en esta técnica se le da un valor que corresponde al peso de importancia de cada parámetro, así de esta manera realiza la eliminación de aquellos cuyo peso es menor.

Para la elección de la cantidad de parámetros en los cálculos mencionados se realizó un cálculo de varianza y se elaboró un gráfico, esto se encuentra dentro del paquete de información que se tiene acceso por medio del manual de usuario anexo

al trabajo; este arrojó que los datos no sufrirían una modificación considerable si se tenían en cuenta un número de 6 variables para la implementación de los modelos.

En cuanto a la validación, se manejaron en los tres modelos la validación cruzada con 5 y 10 validaciones, la cual realiza la segmentación de los datos en conjuntos de prueba y entrenamiento en varias ocasiones, no se realizó una partición fija de los mismos. Para esto se manejó la clase *cross_val_score* de la librería *scikit-learn* la cual se explica mejor en el marco teórico y de manera gráfica en la ilustración 12.

Para la elección de los mejores parámetros dentro de cada modelo de predicción se utilizó '*GridSearchCV',* la cual es una herramienta de '*Scikit-learn'* que permite elegir los que mejor resultado calculen.

Las opciones establecidas para cada parámetro están dadas por el desempeño adecuado en proyectos similares y para evitar el sobre entrenamiento en los modelos.

En el caso del modelo SVM se consideraron los siguientes parámetros:

- C: (0.1, 1, 10, 100).
- Gamma: (1, 0.1, 0.01, 0.001, 0.0001).
- Gamma: (scale, auto).
- Kernel: (linear, poly, rbf, sigmoid).

En el modelo de Redes Neuronales Artificiales (RNA) se contempló lo siguiente:

- Hidden_layer_sizes: ((10,), (20,), (30,), (40,), (50,)).
- Activation: (relu, tanh, logistic).
- Solver: (Adam, lbgs, sgd).
- Alpha: (0.001, 0.05).
- Learning_rate: (constant, adaptative).

Para este modelo se definió un total de 3500 repeticiones frente a los datos, con el fin que no existiera una cantidad insuficiente de predicciones para llegar al punto de convergencia, de esta manera provocando un aprendizaje sin la suficiente cantidad de precisión.

Por último, en el modelo de Bosques Aleatorios se tienen los siguientes parámetros:

- N_estimators: (10, 50, 100, 500).
- Criterion: (gini, entropy).
- Max_depth: (1, 2, 3, 4).
- Min_samples_split: (2, 3, 4, 5, 6).

4. RESULTADOS OBTENIDOS

Con las lecturas anteriores se llegaron a los siguientes resultados por cada método:

- **SVM con PCA**

  Según experimentaciones, se determinó que la mejor combinación de parámetros entre los seleccionados y mencionados en el capítulo anterior son los siguientes:

  - Parametro C: en este parámetro de regularización se determinó que el mejor valor que se toma es 100.
  - Parametro gamma: en este parámetro se determinó que lo mejor es auto.
  - Parametro kernel: en este parámetro se determinó que la mejor opción es rbf.

  Con esta combinación de parámetros se obtuvieron unas medidas aproximadas en las métricas de resultados, como se evidencia en la tabla 7.

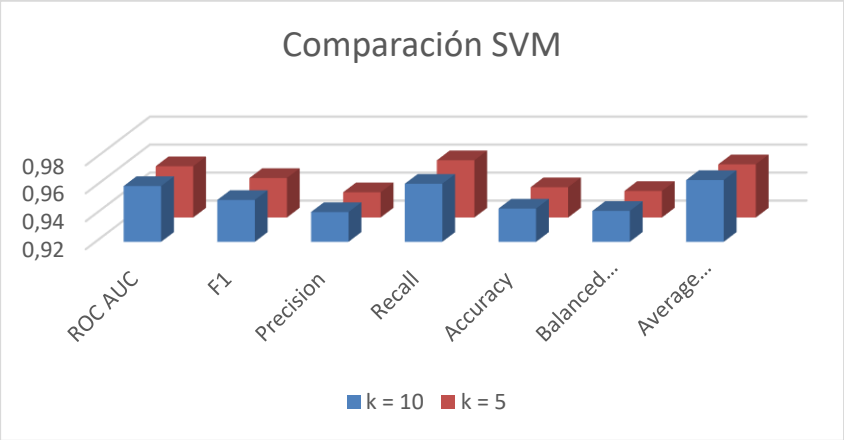
Tabla 7. *Resultados de experimentación con SVM y PCA*

	ROC AUC	F1	Precision	Recall	Accuracy	Balanced accuracy	Average precision
K=10	0.96	0.95	0.9413	0.962	0.9438	0.9421	0.9641
K=5	0.96	0.95	0.9380	0.961	0.9417	0.9390	0.9580

Fuente: Elaboración propia.

En la ilustración 16 se puede notar un mejor desempeño en k = 10, el cual se escogió para compararlo posteriormente con los mejores de los diferentes modelos.

Ilustración 16. *Gráfica de los resultados de experimentación con SVM y*



Fuente: Elaboración propia.

- **Redes Neuronales Artificiales con PCA**

Con base en la experimentación, se estableció que la mejor combinación de parámetros, los cuales se establecieron y mencionaron en el capítulo anterior, son los siguientes:

  - Parámetro Activation: en este parámetro se determinó que la función de activación que mejor se desempeña es tanh.
  - Parámetro Alpha: con los datos proporcionados se estableció que la mejor opción es 0.001.
  - Parámetro hidden_layer_sizes: en este parámetro se determinó que el número de capas ocultas que arroja los mejores resultados son 20.
  - Parámetro Learning_rate: en este parámetro se determinó que la mejor tasa de aprendizaje es constant.
  - Parámetro solver: según la experimentación el método solver que mejor desempeño demuestra es lbfgs.

Con esta combinación de parámetros se obtuvieron medidas aproximadas en las métricas de resultados, como se puede evidenciar en la tabla 8.

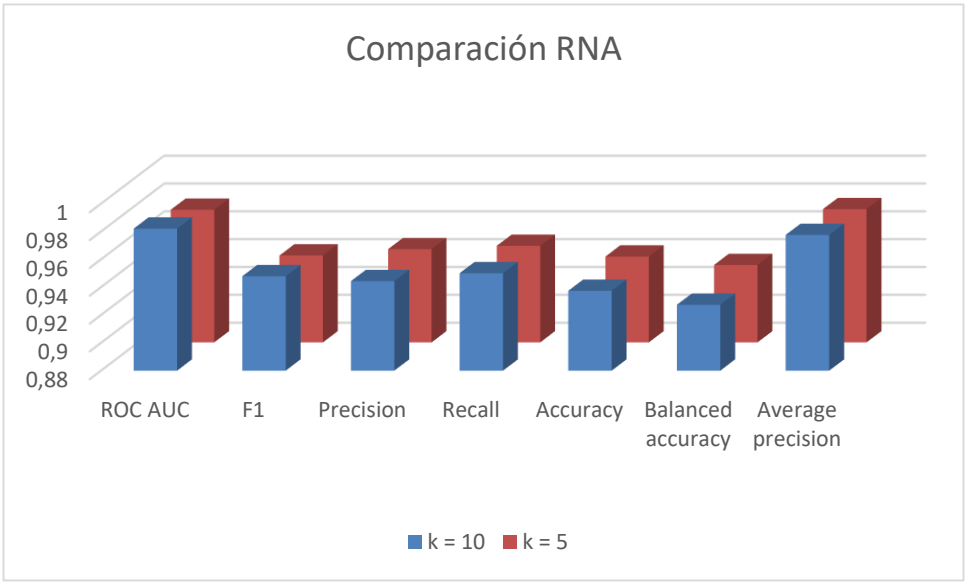
Tabla 8. *Resultados de experimentación con RNA y PCA*

	ROC AUC	F1	Precision	Recall	Accuracy	Balanced accuracy	Average precision
K=10	0.98	0.95	0.9443	0.95	0.9373	0.9273	0.9775
K=5	0.98	0.94	0.9470	0.949	0.9417	0.9356	0.9758

Fuente: Elaboración propia.

En la ilustración 17 se puede notar un mejor desempeño en k = 10, el cual se escogió para posteriormente compararlo con los mejores de los diferentes modelos.

Ilustración 17. *Gráfica de los resultados de experimentación con RNA y PCA*



Fuente: Elaboración propia.

- **Bosques aleatorios con PCA**

Partiendo de las experimentaciones realizadas, se determinó que por cada parámetro, las mejores elecciones fueron:

  - Parámetro criterion: en este parámetro se determinó que la mejor opción es entropy.

- Parámetro max_depth: en este caso se determinó que la profundidad del árbol de decisión debe ser 3.
- Parámetro min_samples_split: para este parámetro se determinó que el número mínimo para la división de nodos es de 2.
- Parámetro n_estimators: se determinó que el número de árboles en el random forest es de 10.

Con esta combinación de parámetros se obtuvieron medidas aproximadas en las métricas de resultados, como se observa en la tabla 9.

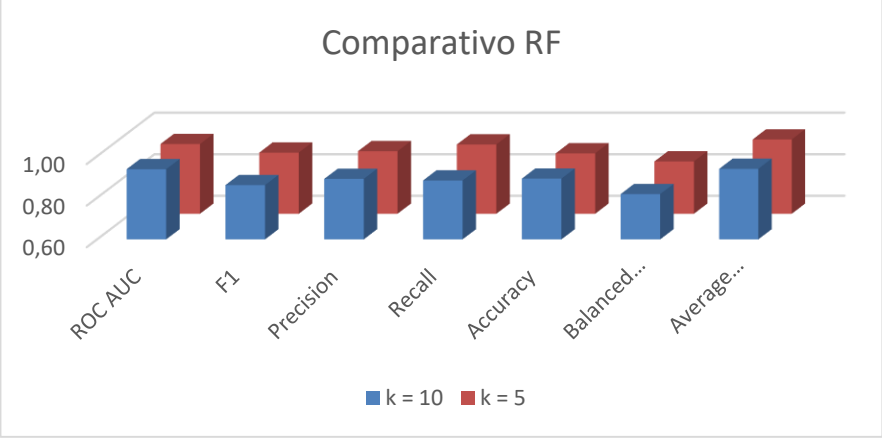
Tabla 9. *Resultados de experimentación con RF y PCA*

	ROC AUC	F1	Precision	Recall	Accuracy	Balanced accuracy	Average precision
K=10	0.94	0.86	0.8911	0.883	0.8922	0.8179	0.9393
K=5	0.94	0.89	0.9010	0.933	0.8906	0.8518	0.9573

Fuente: Elaboración propia.

En la ilustración 18 se puede notar un mejor desempeño en k = 5, el cual se escogió para compararlo posteriormente con los mejores de los diferentes modelos.

Ilustración 18. *Gráfica de los resultados de experimentación con RF y PCA*



Fuente: Elaboración propia.

- **SVM con RFE**

Modificando el método de selección de variables y experimentando con los mismos datos, se determinó que los mejores valores para los siguientes parámetros fueron:

- Parámetro C: en este parámetro de regularización se determinó que el mejor valor que se toma es 10.
- Parámetro gamma: en este parámetro se determinó que lo mejor es auto.
- Parámetro kernel: en este parámetro se determinó que la mejor opción es rbf.

Con esta combinación de parámetros se obtuvieron medidas aproximadas en las métricas de resultados, como se puede evidenciar en la tabla 10.

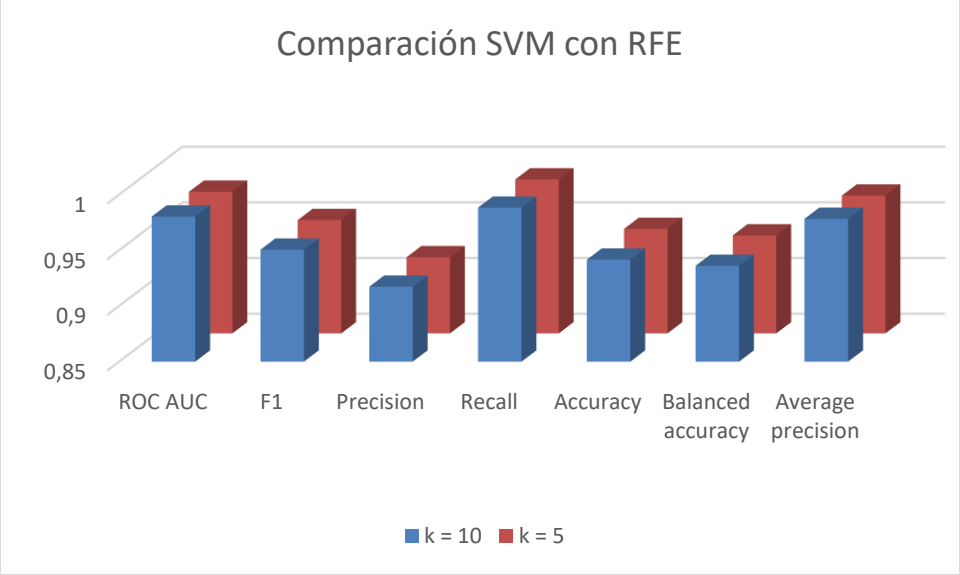
Tabla 10 *Resultados de experimentación con SVM y RFE*

	ROC AUC	F1	Precision	Recall	Accuracy	Balanced accuracy	Average precision
K=10	0.98	0.95	0.9172	0.988	0.9416	0.9361	0.9783
K=5	0.98	0.95	0.9184	0.988	0.9439	0.9381	0.9738

Fuente: Elaboración propia.

Como se evidencia en la ilustración 19, se puede notar un mejor desempeño en k = 5, el cual se escogió para compararlo posteriormente con los mejores de los diferentes modelos.

Ilustración 19. *Gráfica de los resultados de experimentación con SVM y RFE*



Fuente: Elaboración propia.

- **Redes Neuronales Artificiales con RFE**

Con base en la experimentación, realizando el cambio de selección de variables correspondiente, se estableció los mejores valores para los siguientes parámetros:

  - Parametro Activation: en este parámetro se determinó que la función de activación que mejor se desempeña es logistic.
  - Parametro Alpha: con los datos proporcionados se estableció que la mejor opción es 0.001.
  - Parametro hidden_layer_sizes: en este parámetro se determinó que el número de capas ocultas que arroja los mejores resultados son 10.
  - Parametro Learning_rate: en este parámetro se determinó que la mejor tasa de aprendizaje es adaptative.
  - Parametro solver: según la experimentación el método solver que mejor desempeño demuestra es lbfgs.

Con esta combinación de parámetros se obtuvieron estas medidas aproximadas en las métricas de resultados, como se puede evidenciar en la tabla 11.

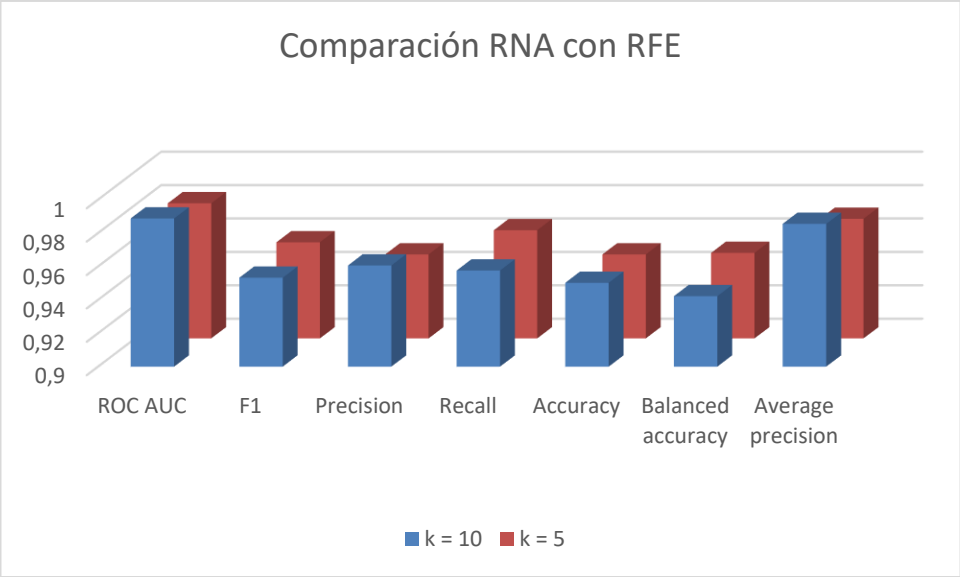
Tabla 11 *Resultados de experimentación con RNA y RFE*

	ROC AUC	F1	Precision	Recall	Accuracy	Balanced accuracy	Average precision
K=10	0.99	0.95	0.9606	0.958	0.9502	0.9422	0.9855
K=5	0.98	0.96	0.9503	0.965	0.9503	0.9512	0.9716

Fuente: Elaboración propia.

Como se evidencia en la ilustración 20, se puede notar un mejor desempeño en k = 10, el cual se escogió para compararlo posteriormente con los mejores de los diferentes modelos.

Ilustración 20. *Gráfica de los resultados de experimentación con RNA y RFE*



Fuente: Elaboración propia.

- **Bosques aleatorios con RFE**

Partiendo de las experimentaciones realizadas aplicando el método de selección de variables RFE, se establecieron por cada parámetro las siguientes elecciones:

- Parámetro criterion: en este parámetro se determinó que la mejor opción es gini.
- Parámetro max_depth: en este caso se determinó que la profundidad del árbol de decisión debe ser 1.
- Parámetro min_samples_split: para este parámetro se determinó que el número mínimo para la división de nodos es de 2.
- Parámetro n_estimators: se determinó que el número de árboles en el random forest es de 10.

Con esta combinación de parámetros se obtuvieron estas medidas aproximadas en las métricas de resultados como se puede evidenciar en la tabla 12.

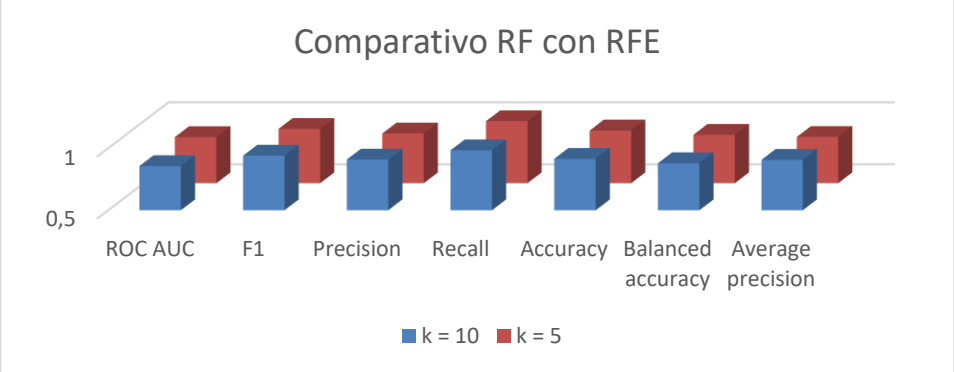
Tabla 12. *Resultados de experimentación con RF y RFE*

	ROC AUC	F1	Precision	Recall	Accuracy	Balanced accuracy	Average precision
K=10	0.86	0.94	0.9089	0.983	0.9144	0.8792	0.9067
K=5	0.98	0.94	0.9008	1.0	0.9251	0.8905	0.8735

Fuente: Elaboración propia.

Como se evidencia en la ilustración 21, se puede notar un mejor desempeño en k = 5, el cual se escogió para compararlo posteriormente con los mejores de los diferentes modelos.

Ilustración 21. *Gráfica de los resultados de experimentación con RF y RFE*



Fuente: Elaboración propia.

Una vez seleccionados las mejores variantes por cada modelo se procedió a realizar una comparativa gráfica de todos para definir el mejor modelo.

La tabla 13 nos muestra todos los puntajes que se seleccionaron para poder realizar la gráfica y proceder a elegir el mejor modelo.

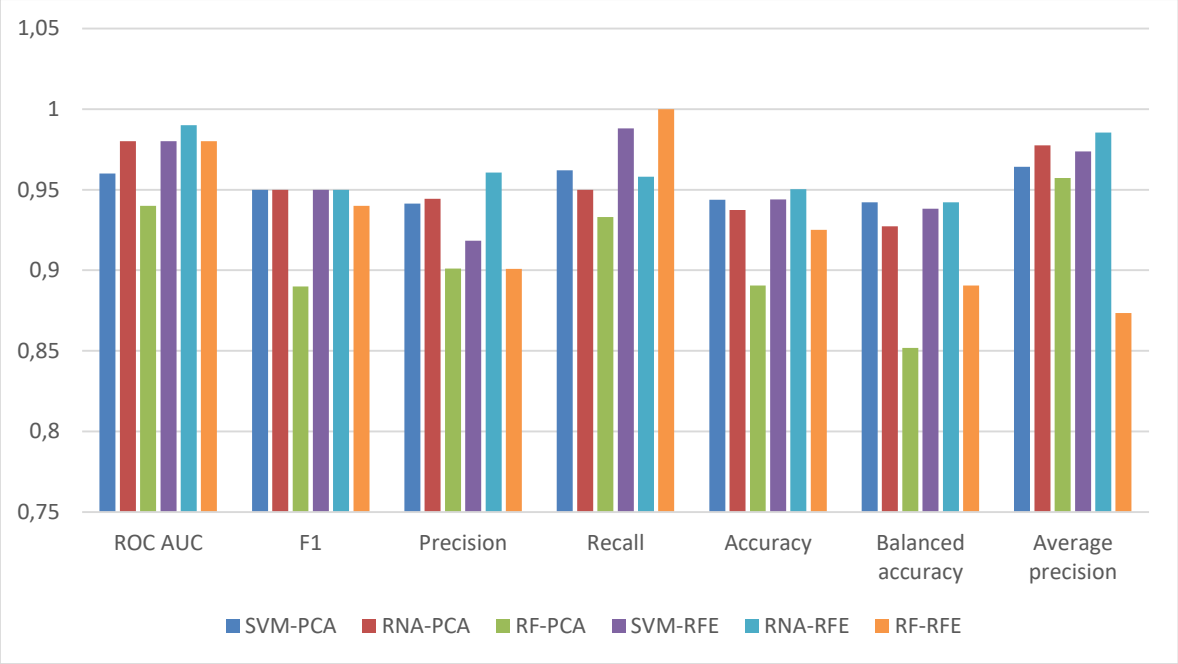
Tabla 13. *Mejores resultados de los modelos*

	ROC AUC	F1	Preci-sion	Recall	Accuracy	Balanced accuracy	Average precision
SVM-PCA	0,96	0,95	0,9413	0,962	0,9438	0,9421	0,9641
RNA-PCA	0,98	0,95	0,9443	0,95	0,9373	0,9273	0,9775
RF-PCA	0,94	0,89	0,901	0,933	0,8906	0,8518	0,9573
SVM-RFE	0,98	0,95	0,9184	0,988	0,9439	0,9381	0,9738
RNA-RFE	0,99	0,95	0,9606	0,958	0,9502	0,9422	0,9855
RF-RFE	0,98	0,94	0,9008	1	0,9251	0,8905	0,8735

Fuente: Elaboración propia.

Según la ilustración 22 se puede deducir que el modelo que mejor se desempeñó en la mayoría de las métricas es el de redes neuronales artificiales (RNA), esto utilizando como método de selección de variables RFE; por lo tanto, será el modelo que se utilizará para realizar el aprendizaje automático.

Ilustración 22. *Gráfica del consolidado de las mejores opciones por modelo*



Fuente: Elaboración propia.

Como se mencionó en un capitulo anterior, los parámetros para la aplicación de este modelo fueron: como activación *logistic,* como valor de alpha 0.001, el número de capas ocultas de 10, una tasa de aprendizaje *adaptive*, y por último un *solver lbfgs.*

5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Los modelos de aprendizaje automático previstos para la realización de este proyecto de grado son los más utilizados para este tipo de problemáticas, aunque algunos de estos tomen cierto tiempo para realizar el proceso, como se evidencio; el rango de precisión incluso en los modelos que no resultaron ser los mejores, es bastante alto, lo cual concluye que se puede llegar a realizar modelos bastante útiles para investigaciones similares a esta de manera satisfactoria.

Este proyecto tiene una gran relevancia para el grupo de investigación LACSER de la Universidad Antonio Nariño y sus avances posteriores a este, ya que es un aporte relevante para el proyecto *"Sistema de comunicación para sobrevivientes de un desastre basado en una red ad hoc de teléfonos",* de igual manera para la sociedad, ya que con base a esta investigación se desprenden varias utilidades que pueden ser acopladas para diferentes fines.

El pre procesamiento de los datos se realizó de manera satisfactoria, brindando así al modelo una información adecuada y veraz acerca de lo recolectado, dando facilidad de entrenamiento a los modelos y mostrando resultados con un desempeño aceptable.

Como se mostró a lo largo del documento, el entrenamiento de los modelos fue uno de los procesos en los que más énfasis se hizo, realizando varías pruebas para llegar a los resultados que, como se evidencian en los resultados fueron de desempeño bastante alto, incluso aquellos que no fueron seleccionados.

Se realizó todo el proceso de entrenamiento de los modelos con una población reducida, esto para efectos de mayor confiabilidad puede ser ampliado en proyectos ajenos a este; el código que se realizó se encuentra capacitado para adaptarse a cualquier cantidad de información suministrada.

Se limitó el número de modelos a los 3 que se consideraban que se acoplaban a lo necesario para concluir satisfactoriamente con lo propuesto al inicio de este trabajo,

sin embargo, se pueden agregar variables o modificarlas, incluso añadir modelos para probar su eficacia en diferentes aspectos.

A pesar del tiempo que se toman algunos de los modelos que se implementaron, no fue necesario el uso de servidores externos o GPUs adicionales para el proceso de aplicación de los modelos; con una máquina con un rendimiento promedio fue suficiente para obtener los resultados.

En la tabla 13 se describe la elección de los mejores modelos por cada método de selección de variables, esto teniendo como opciones la validación cruzada de 5 y 10, teniendo los 6 resultados tabulados se realizó una gráfica que se ve en la ilustración 22, donde de manera más clara y visual se procedió a elegir el modelo de Redes Neuronales Artificiales (RNA) con el método de selección de variables *Recursive Feature Elimination* (RFE).

5.2. Recomendaciones

Si se desea realizar nuevamente el entrenamiento y procesamiento de la información hay que tener en cuenta los tiempos de duración por cada modelo, ya que el proceso total puede emplear un tiempo considerable.

Como se mencionó en el capítulo anterior, el modelo seleccionado fue redes neuronales artificiales con RFE como técnica de selección de variables, el cual es uno de los modelos que más tiempo de implementación emplea; para fines del proyecto no se previó un uso de recursos adicionales debido al tiempo y los recursos. Sin embargo, para fines de utilización final se podría contemplar para optimizar este tiempo y poder tener una buena relación tiempo-eficacia.

Finalmente, se recomienda realizar la lectura del manual relacionado en el anexo 1, con el fin de entender a profundidad la utilización de los modelos; de igual manera en ese documento se encuentra el acceso público a los datos con los cuales se realizó este proyecto para su descarga y posible alteración o comprobación.

6. REFERENCIAS BIBLIOGRÁFICAS

Abad, F. (n.d.). *Curva ROC. La Curva ROC sirve para evaluar la… | by Freddy Abad L. | Medium*. Retrieved September 20, 2020, from https://medium.com/@freddy.abadl/curva-roc-d8c638894f49

Aguado, A. (2016). *Clasificación de actividades humanas en tiempo real a partir de representaciones en esqueleto*. https://addi.ehu.es/handle/10810/19299

Amat, R. J. (2017). *Análisis de Componentes Principales (Principal Component Analysis, PCA) y t-SNE*. RStudio Pubs. https://www.cienciadedatos.net/documentos/35_principal_component_analysis

Amat Rodrigo, J. (2020). *Validación de modelos predictivos (machine learning): Cross-validation, OneLeaveOut, Bootstraping*. Cienciadedatos. https://www.cienciadedatos.net/documentos/30_cross-validation_oneleaveout_bootstrap

Anguita, D., Ghio, A., Oneto, L., Parra, X., & Reyes-Ortiz, J. L. (2013). A public domain dataset for human activity recognition using smartphones. *ESANN 2013 Proceedings, 21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*.

Balagueró, T. (2018). *¿Qué son los datasets y los dataframes en el Big Data? | Deusto Formación*. https://www.deustoformacion.com/blog/programacion-diseno-web/que-son-datasets-dataframes-big-data

Barrios, J. (2019). *La matriz de confusión y sus métricas – Inteligencia Artificial –*. Health Big Data. https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/

Batiste, A. (2011). *Orígenes de las redes mesh I: las primeras redes ad-hoc*. https://sevillamesh.wordpress.com/2011/02/22/origenes-de-las-redes-mesh-i-las-primeras-redes-ad-hoc/

Briega, R. E. L. (n.d.). *Introducción a la inteligencia artificial*. Retrieved September 7, 2020, from https://relopezbriega.github.io/blog/2017/06/05/introduccion-a-la-inteligencia-artificial/

Buran. (2004). Redes ad-hoc : el próximo reto. *Buran*.

Caicedo, B., & López, J. (2010). Redes Neuronales Artificiales. In *Charlas de fisica*. https://doi.org/10.1016/S0210-5691(05)74198-X

Calvo, D. (2017). *Clasificación de redes neuronales artificiales - Diego Calvo*. https://www.diegocalvo.es/clasificacion-de-redes-neuronales-artificiales/

Capacci, A., & Mangano, S. (2015). Las catástrofes naturales. *Cuadernos de Geografía: Revista Colombiana de Geografía*, *24*(2), 35–51. https://doi.org/10.15446/rcdg.v24n2.50206

Caparrini, F. S. (2019). *Aprendizaje Inductivo: Árboles de Decisión - Fernando Sancho Caparrini*. http://www.cs.us.es/~fsancho/?e=104

CleverData. (2019). *¿Qué es Machine Learning? – Cleverdata*. ¿Que Es Machine Learning?

Colombia, M. de E. N. de. (n.d.). *Habeas Data - Ministerio de Educación Nacional de Colombia*. Retrieved September 10, 2020, from https://www.mineducacion.gov.co/1759/w3-article-387771.html?_noredirect=1

Cuenca, U. politécnica salesiana sede. (2012). *Universidad politécnica salesiana sede cuenca "análisis de la propuesta de evolución de redes 3g y su convergencia a la tecnología 4g para redes de*. http://dspace.ups.edu.ec/handle/123456789/2072

DataTask. (n.d.). *Hyperparameters and Parameters*. Retrieved April 11, 2021, from https://kola40.com/2021/01/18/hyperparameters-and-parameters/

Fundamentos Básicos. (n.d.). Retrieved September 8, 2020, from http://www.varpa.org/~mgpenedo/cursos/scx/Tema1/nodo1-2.html

Girault, J. A. (2013). Las Neuronas. *Cuadernos M y C*, *4*, 82–87. https://doi.org/10.1371/journal.pbio.0040311

Heras, J. M. (n.d.). *Precision, Recall, F1, Accuracy en clasificación - IArtificial.net*. Retrieved September 20, 2020, from https://www.iartificial.net/precision-recall-f1-accuracy-en-clasificacion/

Hormazábal, C. (2019). Comunicación e imagen en crisis: análisis de empresas de telecomunicaciones en Chile tras el 27F de 2010. *Cuadernos Del Centro de*

Estudios de Diseño y Comunicación. https://doi.org/10.18682/cdc.v40i40.1440

Https://scikit-learn.org/. (n.d.). *sklearn.preprocessing.StandardScaler — scikit-learn 0.24.2 documentation*. Retrieved May 26, 2021, from https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

InteractiveChaos. (n.d.). *Matriz de confusión | Interactive Chaos*. Retrieved April 18, 2021, from https://interactivechaos.com/es/manual/tutorial-de-machine-learning/matriz-de-confusion

Kashyap, K. (n.d.). *Machine Learning- Decision Trees and Random Forest Classifiers | by Karan Kashyap | Analytics Vidhya | Medium*. Retrieved September 20, 2020, from https://medium.com/analytics-vidhya/machine-learning-decision-trees-and-random-forest-classifiers-81422887a544

López, A., Villarreal, E., & Álvarez, C. I. (2016). Migración de fuentes sísmicas a lo largo del cinturón de fuego del pacífico. *La Granja*, *25*(1), 5. https://doi.org/10.17163/lgr.n25.2017.01

Martínez, J. (2020). *Precision, Recall, F1, Accuracy en clasificación - IArtificial.net*. https://www.iartificial.net/precision-recall-f1-accuracy-en-clasificacion/

McClelland, S. (2011). *Japan: Surviving a tsunami, rebuilding communications: Biblioteca Virtual de la Pontificia Universidad Católica del Perú*. Intermedia (0309118X). Dec2011, Vol. 39 Issue 5, P12-21. 8p.

Medina-Merino, R. F., & Ñique-Chacón, C. I. (2017). Bosques aleatorios como extensión de los árboles de clasificación con los programas R y Python. *Interfases*. https://doi.org/10.26439/interfases2017.n10.1775

Medium. (2019). *An introduction to Grid Search*. Data Driven Investor. https://www.mygreatlearning.com/blog/gridsearchcv/#sh1

Na8. (2018). *Comprende Principal Component Analysis | Aprende Machine Learning*. https://www.aprendemachinelearning.com/comprende-principal-component-analysis/

Oberto, C. (n.d.). *Curva Operador-Receptor (ROC) – Página en construcción*. Retrieved November 8, 2020, from

https://modulodeestadistica.wordpress.com/curva-operador-receptor-roc/

Oliveira García-Ollala, O. (2019). *Redes Neuronales artificiales: Qué son y cómo se entrenan*. Xeridia. https://www.xeridia.com/blog/redes-neuronales-artificiales-que-son-y-como-se-entrenan-parte-i

Organización de las Naciones Unidas para la Educación la Ciencia y la Cultura. (1952). *Convención Universal sobre Derecho de Autor*. Organización de Las Naciones Unidas Para La Educación La Ciencia y La Cultura. http://portal.unesco.org/es/ev.php-URL_ID=15381&URL_DO=DO_TOPIC&URL_SECTION=201.html

Python Software Foundation. (2016). *Python Documentation: History and License*. https://docs.python.org/3/license.html

Redacción APD. (2019). *Qué es Machine Learning, cómo funciona y a qué se aplica*. https://www.apd.es/que-es-machine-learning/

República, C. de la. (n.d.). *CONGRESO DE LA REPÚBLICA*.

Rodrigo, J. (2017). Máquinas de Vector Soporte (Support Vector Machines, SVMs). *Cienciadedatos*. https://www.cienciadedatos.net/documentos/34_maquinas_de_vector_soporte_support_vector_machines

Scientia Et Technica. (2005). LAS MÁQUINAS DE SOPORTE VECTORIAL (SVMs). *Scientia Et Technica*. https://doi.org/10.22517/23447214.6895

Shinmoto Torres, R. L., Ranasinghe, D. C., Shi, Q., & Sample, A. P. (2013). Sensor enabled wearable RFID technology for mitigating the risk of falls near beds. *2013 IEEE International Conference on RFID, RFID 2013*, 191–198. https://doi.org/10.1109/RFID.2013.6548154

Sitiobigdata. (2019). *Machine Learning: Seleccion Metricas de clasificacion*. https://sitiobigdata.com/2019/01/19/machine-learning-metrica-clasificacion-parte-3/

Weiss, G. M., Yoneda, K., & Hayajneh, T. (2019). Smartphone and Smartwatch-Based Biometrics Using Activities of Daily Living. *IEEE Access*, *7*, 133190–133202. https://doi.org/10.1109/ACCESS.2019.2940729

Wikipedia, la enciclopedia libre. (n.d.). *Validación cruzada - Wikipedia, la enciclopedia libre*. Retrieved April 6, 2021, from https://es.wikipedia.org/wiki/Validación_cruzada

Yellowbrick. (n.d.). *Recursive Feature Elimination — Yellowbrick*. Retrieved March 30, 2021, from https://www.scikit-yb.org/en/latest/api/model_selection/rfecv.html