

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

FACULTAD DE INGENIERÍA

Proyecto Curricular de Ingeniería Electrónica

ANÁLISIS DE SEÑALES ELECTROCARDIOGRÁFICAS PARA LA DETECCIÓN TEMPRANA DEL SÍNDROME METABÓLICO MEDIANTE MACHINE LEARNING

Monografía como requisito para optar por el título de:
Ingeniero Electrónico

Juan Andrés Peláez Grajales
Código: 20181005027
Santiago Arzuaga Robles
Código: 20182005057

Director: Edmundo Vega Osorio.

Grupo de investigación:
Bionanotecnología

Bogotá D.C, Colombia.
Enero, 2025

Resumen

En el presente trabajo se busca desarrollar un método innovador para la identificación precoz del síndrome metabólico (SM) a través del estudio de señales electrocardiográficas (ECG) empleando técnicas avanzadas de aprendizaje automático. El SM, identificado por una serie de irregularidades metabólicas que aumentan considerablemente el riesgo de padecer enfermedades cardiovasculares, diabetes tipo 2 y accidentes cerebrovasculares, se diagnostica hoy en día mediante métodos invasivos y tediosos para los pacientes, como lo es la prueba oral de tolerancia a la glucosa. Este proyecto tiene como objetivo reemplazar estas prácticas por un método no invasivo, ágil y eficaz fundamentado en el análisis automatizado de señales ECG, teniendo en cuenta que el SM influye en la modulación autónoma del corazón, lo que resulta en alteraciones perceptibles en el ECG.

El estudio abarca la creación de un sistema que procesa señales ECG, suprimiendo el ruido y obteniendo parámetros relevantes en los dominios del tiempo, la frecuencia y no lineales. Estos parámetros se emplean en el entrenamiento y prueba de dos modelos de aprendizaje automático: Random Forest y Support Vector Machine (SVM).

Los resultados obtenidos resaltan el potencial del análisis de ECG como una herramienta diagnóstica no invasiva. Además, la implementación de algoritmos de aprendizaje automático ha demostrado ser una solución eficiente para la detección automatizada del SM, con aplicaciones que podrían extenderse a otras patologías relacionadas con la modulación autónoma cardíaca.

Dedicatoria

Juan Andrés Peláez Grajales

A mi familia, en especial a mi mamá Claudia Grajales, a mis abuelos, Mario Grajales y María Elena Echeverry, a mi tía Eliana Grajales, por todo el amor, apoyo incondicional y acompañamiento que me han brindado durante este arduo, pero enriquecedor proceso, por siempre confiar en mis capacidades, por ser mi motivación en los momentos difíciles y por enseñarme tantos valores valiosos para la vida. También una mención especial para aquellas personas que compartieron este camino conmigo y a quienes vieron mi crecimiento personal y profesional a lo largo de estos años.

Santiago Arzuaga Robles

A mis padres, María Teresa y Luis Erasmo, por su amor incondicional, su incansable apoyo y por enseñarme con su ejemplo el valor del esfuerzo, la perseverancia y los sueños cumplidos. Gracias por ser mi mayor inspiración y mi fortaleza en cada paso de este camino. A mis hermanos, Valeria y Samuel, por ser mis compañeros de vida, mis cómplices y mi constante motivo de alegría.

Agradecimientos

En primer lugar, nos gustaría agradecer al profesor Edmundo Vega Osorio, quien, con su dedicación, experiencia, orientación y valiosas sugerencias, nos guió a lo largo del proceso de investigación y redacción de este libro. Su guía constante nos motivó a alcanzar niveles que nunca imaginamos.

Finalmente, deseamos extender nuestro reconocimiento a los docentes de la Universidad Distrital Francisco José de Caldas, quienes nos abrieron las puertas y nos brindaron las oportunidades de avanzar en nuestra carrera profesional, compartieron su sabiduría y conocimiento, haciéndonos crecer tanto personal como profesionalmente.

Índice General

Resumen **1**

Dedicatoria **2**

Agradecimientos **3**

Acrónimos **7**

Índice de figuras **9**

Índice de tablas **10**

1. Introducción al proyecto **12**
1.1. Formulación del problema 12
1.2. Justificación 14
1.3. Objetivo General 16
1.4. Objetivos Específicos 16
1.5. Alcances y limitaciones 17

2. Fundamentos Teóricos del Síndrome Metabólico **18**
2.1. Generalidades 18
2.2. Epidemiología 19
2.3. ¿A quiénes y cómo evaluar el síndrome metabólico? 19
2.4. Prueba oral de tolerancia a la glucosa 21

3. Fundamentos Teóricos de señales electrocardiográficas (ECG) **22**
3.1. Generalidades 22
3.2. Componentes tradicionales del ECG 23
3.2.1. Onda P 23
3.2.2. Intervalo PR 24
3.2.3. Complejo QRS 24

3.2.4. Intervalo QTc . . . 25
3.2.5. Segmento ST . . . 25
3.2.6. Onda T . . . 25

4. Fundamentos Teóricos de filtrado de señales 26
4.1. Generalidades . . . 26
4.2. Filtros digitales . . . 27
4.3. Filtros Ideales . . . 27
4.4. Tipo de filtros . . . 28
4.4.1. Filtros pasabajos . . . 28
4.4.2. Filtros pasaaltos . . . 28
4.4.3. Filtros pasabanda . . . 29
4.4.4. Filtro Butterworth . . . 30

5. Fundamentos Teóricos de Machine Learning 32
5.1. Generalidades . . . 32
5.2. El aprendizaje de un sistema de Machine Learning . . . 33
5.2.1. Tipos de modelos de Machine Learning . . . 33
5.3. Aplicaciones del aprendizaje supervisado . . . 38
5.4. Preprocesamiento de datos . . . 39
5.4.1. Conceptos estadísticos para el preprocesamiento de datos . . . 40

6. Estado del arte 43

7. Desarrollo de Ingeniería 45
7.1. Análisis estadístico del conjunto de datos . . . 45
7.1.1. Prueba de Kolmogorov Smirnov . . . 49
7.1.2. Prueba U de Mann-Whitney . . . 56
7.1.3. Comparación de la mediana en los grupos de estudio . . . 60
7.2. Preprocesamiento de las señales ECG . . . 65
7.2.1. Extracción de las señales ECG sin filtrar . . . 66
7.2.2. Primer filtrado de las señales ECG . . . 66
7.2.3. Segundo filtrado de las señales ECG . . . 67
7.2.4. Detección de picos y segmentos de las señales ECG . . . 68
7.3. Series de tiempo . . . 70
7.3.1. Series de tiempo intervalos RR . . . 71
7.3.2. Series de tiempo intervalos PR . . . 73
7.3.3. Series de tiempo intervalos QTc . . . 76
7.3.4. Series de tiempo segmentos ST . . . 78
7.4. Parámetros de la señal ECG . . . 81

7.4.1. Parámetros en el dominio del tiempo 81
7.4.2. Parámetros en el dominio de la frecuencia 86
7.4.3. Parámetros no lineales 89
7.5. Machine Learning 91
7.5.1. Random Forest 91
7.5.2. Support Vector Machine 101

8. Resultados **107**
8.1. Random Forest 107
8.2. Support Vector Machine 110
8.3. Comparación de los modelos RF vs SVM 113
8.3.1. Resultados generales 113
8.3.2. Recursos computacionales consumidos 115

9. Conclusiones **117**

Referencias **119**

Anexos **127**
9.1. Código en Python del modelo ML de Random Forest 127
9.2. Código en Python del modelo ML de Support Vector Machine 130

Acrónimos

SM Síndrome Metabólico.

ML Machine Learning.

ECG Electrocardiográficas.

OMS Organización Mundial de la Salud.

DANE Departamento Administrativo Nacional de Estadística.

PTGO Prueba de Tolerancia a la Glucosa Oral.

VFC Variabilidad de la Frecuencia Cardíaca.

ECV Enfermedad Cardiovascular.

IMC Índice de Masa Corporal.

ADA American Diabetes Association.

FIR Filtro de Respuesta Finita al Impulso.

IIR Filtro de Respuesta Infinita al Impulso.

SVM Support Vector Machine.

RF Random Forest.

KS Prueba de Kolmogorov Smirnov.

SDNN Desviación estándar de los intervalos RR normales.

HR Frecuencia cardíaca promedio.

STD HR Desviación estándar del ritmo cardíaco.

SVC Support Vector Classifier.

Índice de figuras

1.1. Mayores causas de muerte en el año 2021, 2022 y parte del 2023 [5]. 13

3.1. Componentes de una señal electrocardiográfica [22]. 23

4.1. Filtrado de señales [24]. 27
4.2. Respuesta filtro pasa bajos. 28
4.3. Respuesta filtro pasaaltos. 29
4.4. Respuesta filtro pasabanda. 30
4.5. Diagrama de Bode del filtro Butterworth [27]. 31

5.1. Representación gráfica de una Máquina de Vectores de Soporte. 35
5.2. Separación lineal de los datos en SVM. 36
5.3. Separación no lineal de los datos en SVM. 37
5.4. Estructura de un árbol de decisión (RF). 38
5.5. Etapas del preprocesamiento de datos [32]. 39
5.6. Ejemplo prueba de Kolmogórov-Smirnov [34]. 42

7.1. Análisis estadístico en los grupos de estudio. 48
7.2. Gráficas presentadas en valores de la mediana para seis características. . . 62
7.3. Gráficas presentadas en los valores de la mediana para cinco características. 63
7.4. Gráfica de la glucosa en diferentes tiempos en valores de la mediana. 64
7.5. Gráfica de la insulina en diferentes tiempos en valores de la mediana. . . . 64
7.6. Señales ECG sin filtrar correspondientes a la segunda derivación 66
7.7. Señales ECG filtradas por primera vez correspondientes a la segunda derivación 67
7.8. Señales ECG filtradas por segunda vez correspondientes a la segunda derivación 68
7.9. Detección de picos y segmentos en la segunda derivación de las señales ECG. 69
7.10. Intervalo RR señal ECG [64]. 71
7.11. Series de tiempo del intervalo RR en diferentes momentos. 73

7.12. Intervalo PR señal ECG [67]. 74
7.13. Series de tiempo del intervalo PR en diferentes momentos. 75
7.14. Intervalo QTc señal ECG [70]. 76
7.15. Series de tiempo del intervalo QTc en diferentes momentos. 77
7.16. Intervalo ST señal ECG [67]. 78
7.17. Series de tiempo del intervalo ST en diferentes momentos. 79
7.18. Desviación del segmento ST durante el proceso del POTG. 80
7.19. Predicción RF ejemplo [77]. 93
7.20. Validación cruzada vs Número de árboles. 96
7.21. Validación cruzada vs Número de predictores. 96
7.22. Out-of-Bag error vs Número de árboles. 97
7.23. Out-of-Bag error vs Número de predictores. 97
7.24. Valores de random state y exactitud del modelo. 99
7.25. Separación de las observaciones haciendo aumento de los espacios [78] . . . 102
7.26. Valores de random state y exactitud del modelo. 105

8.1. Resultado de la carga y conjunto de datos del modelo RF. 107
8.2. Predicción del modelo, resultados esperados y exactitud de este mismo. . . 108
8.3. Promedio de las exactitudes del modelo dependiendo del random_state desde 0 hasta 100. 109
8.4. Recursos computacionales utilizados por el método RF. 109
8.5. Resultado de la carga y conjunto de datos del modelo SVM. 110
8.6. Predicción del modelo, resultados esperados y exactitud de este mismo. . . 111
8.7. Promedio de las exactitudes del modelo dependiendo del random_state desde 0 hasta 100. 111
8.8. Recursos computacionales utilizados por el método SVM. 112
8.9. Comparación de las exactitudes de los modelos. 114

Índice de tablas

7.1. Conjunto de datos IMC. 49
7.2. Percentil conjunto de datos IMC. 50
7.3. Valor tipificado conjunto de datos IMC 51
7.4. Probabilidad de Z_i. 52
7.5. Valor absoluto de la diferencia entre la probabilidad y el percentil. 53
7.6. Valor absoluto del dato anterior 54
7.7. Valores de Kolmogorov Smirnov 54
7.8. Valores del colesterol de los grupos de estudio 56
7.9. Conjunto de datos y su rango 57
7.10. Características de los grupos de estudio, valores dados en mediana. 61
7.11. Valores normales en latidos por minuto y segundos para el intervalo RR [65]. 72
7.12. Parámetros en el Dominio del Tiempo de la VFC [72], [73]. 85
7.13. Parámetros en el Dominio de la Frecuencia de la VFC [74], [75]. 88
7.14. Parámetros No Lineales de la VFC [72], [73]. 90
7.15. Ejemplo de predicción [77] 92

8.1. Exactitud promedio de los dos modelos de ML. 113
8.2. Recursos computacionales consumidos por cada uno de los métodos. 115

Capítulo 1

Introducción al proyecto

1.1. Formulación del problema

El síndrome metabólico (SM) son un conjunto de anomalías metabólicas, las cuales son consideradas factores de riesgo en un mismo individuo, para desarrollar diabetes y enfermedades cardiovasculares [1]. Siendo estas, según la Organización Mundial de la Salud (OMS), tres de las cinco enfermedades que más muertes generó en el año 2019, donde, junto al cáncer de pulmón y la neumopatía obstructiva crónica, provocaron 100 millones más de años de vida saludable perdidos a comparación del año 2000 [2].

Se conoce que la prevalencia del SM, es decir, "la proporción de casos de una enfermedad en un periodo de tiempo, respecto a la población existente en la zona objeto de estudio" [3]; varía según las diferentes características que presenta cada individuo, bien sea su edad, sexo, origen étnico y estilo de vida [1]. La OMS ha determinado que dicha prevalencia, para la población adulta en el mundo, se estima entre el 20 % y 25 % de padecimiento del SM. Ocasionando así, el incremento de las probabilidades de morir hasta dos veces más y triplicando el riesgo de sufrir un ataque cardíaco o un accidente cerebrovascular en comparación con personas las cuales no presentan esta serie de anomalías o alteraciones [1]. El porcentaje de la prevalencia del SM, aumenta a medida que las personas se hacen mayores, con 20 años de edad se tiene un registro del 24 %, con personas mayores de 50 años de edad se habla de un 30 % o más y para aquellos mayores de 60 años de edad se conoce que hay más del 40 % de prevalencia [1].

Un estudio realizado entre los años 2014 y 2017, en Colombia, que incluyó a jóvenes de Bogotá, Boyacá y Cali, dio como resultado una prevalencia del 16.4 %, estando presente está en un 11.1 % en hombre y el 5.3 % en mujeres. En otra parte del territorio nacional, como lo es Medellín y sus alrededores, por medio de un estudio realizado en estos mismos

años, se obtuvo que la prevalencia del SM en adultos es cercana al 40 % [4].

Según cifras del Departamento Administrativo Nacional de Estadística (DANE), entre los años 2021, 2022 y los diez primeros meses del 2023, tres de las cinco mayores causas de defunción en Colombia están relacionadas con enfermedades cardiovasculares. Teniendo las siguientes cifras mostradas en la figura 1.1 [5].

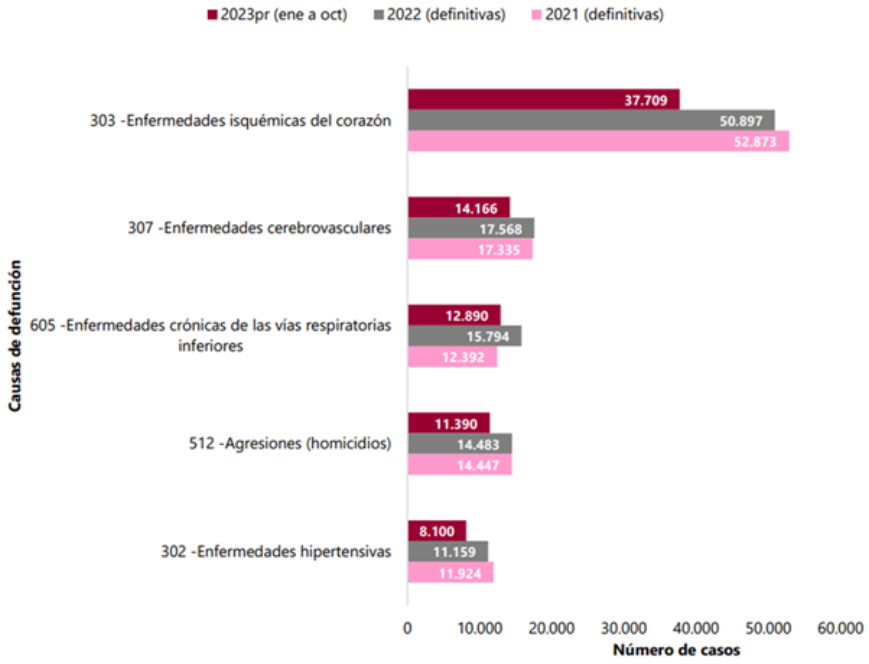


Figura 1.1: Mayores causas de muerte en el año 2021, 2022 y parte del 2023 [5].

Se conoce de antemano que la práctica de la medicina, tiene un gran factor humano involucrado en cada uno de los procesos que se desarrollan, esto ocasiona que aumente aún más las posibilidades de obtener, como paciente, un diagnóstico errado. Un estudio realizado en el hospital San Ignacio, reveló que el 37 % de las historias clínicas realizadas presentaban uno o más errores, obteniendo un total de 59 errores, en donde se promedian 1.6 errores por historias clínicas, siendo 100 los archivos analizados [6].

El procedimiento para la correcta detección del SM consiste en realizar un proceso de aproximadamente 3 horas, en donde el paciente estará siendo sometido a extracciones de sangre (la primera en ayunas) cada 30 a 60 minutos, después de haber ingerido un líquido que contiene, generalmente, 75 gramos de glucosa. Este procedimiento se denomina prueba

de tolerancia a la glucosa oral (PTGO) [7], el cual presenta una pequeña probabilidad de sufrir efectos secundarios como náuseas, desmayos, dificultad para respirar, sudoración o mareos. Como se puede ver, es una prueba altamente invasiva en donde el paciente está constantemente pasando por momentos los cuales, para muchas personas no son nada agradables, no solo siendo eso, también se tiene la probabilidad de sufrir efectos secundarios que, aunque sea baja no llegan a ser inexistentes dichos efectos. Si a esto se le agrega el tiempo el cual las personas están durante dicho examen, se observa que es un proceso para diagnóstico invasivo, tedioso y poco amigable con los pacientes.

Por estos motivos anteriormente expuestos, desde el grupo de investigación de Bionanotecnología, se ha venido trabajando en el presente proyecto el cual busca proponer una solución basada en la detección del SM por medio de las señales electrocardiográficas (ECG), con el fin de disminuir la posibilidad de sufrir un diagnóstico errado de su condición, además, se pretende brindar una solución más amigable con el paciente, donde no pasará por procesos invasivos y no tendrá largas horas de espera en procedimientos. Se menciona que el procedimiento no será invasivo, ya que, se logrará obtener un diagnóstico por medio del procedimiento habitual efectuado en un examen electrocardiográfico, este resultado se elaborará gracias a que el SM deteriora la modulación autónoma cardiaca, aumentando la frecuencia cardiaca y la presión arterial, y disminuyendo la variabilidad de la frecuencia cardiaca [8], lo cual se podrá apreciar por medio de una anomalía en el ECG.

¿Será posible analizar los parámetros presentes en las señales ECG mediante el uso de técnicas de procesamiento digital de señales para el diagnóstico temprano del síndrome metabólico?

1.2. Justificación

El síndrome metabólico (SM) es un conjunto de trastornos metabólicos que aumentan el riesgo de desarrollar enfermedades cardíacas, accidentes cerebrovasculares, diabetes tipo 2 y otras enfermedades crónicas. La detección temprana del SM es importante para prevenir estas complicaciones.

Actualmente, el diagnóstico del SM se basa en un conjunto de criterios clínicos que incluyen la medición de la circunferencia de la cintura, los niveles de glucosa, colesterol y triglicéridos en sangre. Estos criterios pueden ser costosos, laboriosos y subjetivos. El análisis de electrocardiogramas (ECG) para la detección temprana del SM es una nueva técnica que tiene el potencial de ser más precisa, rentable y fácil de usar que los métodos actuales.

Este proyecto de investigación se erige como un pilar fundamental para ampliar y enriquecer el conocimiento científico que tenemos sobre el síndrome metabólico (SM) y, en particular, sobre los métodos de detección temprana de esta compleja condición. El análisis de electrocardiogramas (ECG) como herramienta de detección abre un horizonte innovador y prometedor que podría trascender los límites de los métodos convencionales.

En términos económicos, la implementación exitosa de este enfoque podría traducirse en beneficios tangibles. La prevención de enfermedades crónicas asociadas al síndrome metabólico no solo incide directamente en la calidad de vida de los individuos, sino que también conlleva un impacto económico significativo. Las enfermedades crónicas, como las cardiovasculares y la diabetes tipo 2, imponen una carga financiera considerable en los sistemas de salud. La detección temprana, al evitar o retrasar el desarrollo de estas enfermedades, podría resultar en ahorros sustanciales en costos de tratamiento y gestión de enfermedades a largo plazo.

Un aspecto crucial para este proyecto es la implementación de un algoritmo de aprendizaje automático para analizar los ECG, lo cual no solo aumenta la precisión del diagnóstico, sino que también conlleva una significativa reducción de costos. La automatización de procesos mediante software especializado no solo agiliza el proceso de diagnóstico, sino que minimiza la necesidad de recursos humanos y materiales asociados con los métodos tradicionales.

Desde la perspectiva social, la detección temprana del síndrome metabólico (SM) emerge como una herramienta fundamental para mejorar la calidad de vida de las personas y aliviar la carga de enfermedades (ECG), adquiere un papel trascendental en este escenario. Al democratizar el acceso a una detección temprana más efectiva, se establece un puente que abarca todas las clases sociales, permitiendo que individuos de diversos contextos económicos se beneficien por igual.

La equidad en la detección temprana del SM no solo tiene un impacto inmediato en la salud individual, sino que también repercute en la salud colectiva. La prevención y el tratamiento oportuno derivados de una detección temprana pueden reducir la carga de enfermedades en la sociedad en su conjunto. Esto se traduce en un alivio para los sistemas de salud, liberando recursos para atender otras necesidades médicas y mejorando la eficiencia en la gestión de la salud pública.

Desde la perspectiva ambiental, el proyecto también presenta una dimensión sostenible y responsable. La reducción del uso de pruebas clínicas tradicionales, con la consiguiente disminución de la producción y desecho de materiales asociados, contribuye significativa-

mente a la reducción de la contaminación ambiental. Los métodos más eficientes, como el análisis de ECG, se destacan no solo por su precisión diagnóstica sino también por su menor impacto ecológico. Esto alinea el proyecto con una visión más sostenible de la medicina y la investigación médica, enfocada en minimizar la huella ambiental de las prácticas clínicas.

Este proyecto no solo tiene un impacto positivo en el ámbito de la salud, sino que también destaca la importancia de la ingeniería electrónica en la creación de soluciones innovadoras para problemas médicos complejos. Al combinar hardware de precisión con algoritmos avanzados de machine learning, se puede ofrecer una herramienta poderosa y eficiente para la detección temprana de enfermedades cardiovasculares.

1.3. Objetivo General

Analizar los parámetros presentes en las señales ECG para el diagnóstico temprano del síndrome metabólico, haciendo uso de técnicas de machine learning implementadas por medio de Python.

1.4. Objetivos Específicos

- Desarrollar un sistema de filtrado de señales ECG, con el propósito de aislar de manera efectiva las señales cardíacas de interés y eliminar el ruido proveniente de otras fuentes.
- Extraer los parámetros de la variabilidad de la frecuencia cardiaca (VFC) de las señales en el dominio del tiempo y la frecuencia de los registros electrocardiográficos de los grupos de estudio, un grupo que padecen del síndrome metabólico y un grupo sano.
- Aplicar dos métodos de aprendizaje automático en los parámetros obtenidos de las señales ECG de cada grupo para encontrar patrones capaces de identificar problemas metabólicos.
- Recopilar y analizar de los resultados producidos por cada método, con el fin de realizar una comparación detallada de los resultados obtenidos de ambos métodos y así determinar cuál de los dos métodos proporciona mejores resultados en términos de precisión, eficiencia y fiabilidad.

1.5. Alcances y limitaciones

El proyecto de grado se enfoca en desarrollar un software que pueda leer señales electrocardiográficas (ECG) y extraer parámetros importantes para realizar un diagnóstico. Es necesario emplear dos enfoques distintos de aprendizaje automático para analizar estos datos y luego compararlos con el fin de identificar cuál es más eficaz para este tipo de análisis. Esperamos que el software pueda automatizar parte del proceso de diagnóstico electrocardiográfico, lo que permitiría una evaluación preliminar de las señales ECG. Se utilizarán técnicas de procesamiento digital de señales y herramientas de inteligencia computacional para alcanzar este objetivo. El proyecto utilizará una metodología que consta de la recopilación de datos, el preprocesamiento de las señales ECG y el entrenamiento de modelos de machine learning con el fin de llevar a cabo el análisis y la comparación entre ellos.

En lo que respecta a las limitaciones, hay varios desafíos en el proyecto que podrían impactar su precisión y aplicabilidad. La falta de validación por parte de un cardiólogo es uno de los principales obstáculos. Debido a la dificultad para contar con la colaboración de un profesional médico experimentado en cardiología, el proyecto no puede asegurar que sus resultados sean completamente precisos desde una perspectiva clínica. El software debe ser considerado como una herramienta auxiliar y no como un sustituto del diagnóstico médico profesional, debido a esta limitación. El proyecto se ve restringido por la calidad y cantidad de datos disponibles para entrenar y validar los modelos de aprendizaje automático. Esto se debe a que las señales provienen de una base de datos existente, sin posibilidad de añadir información sobre nuevos pacientes. Esta limitación puede afectar tanto el rendimiento como la aplicabilidad del software en situaciones del mundo real.

Resumiendo, el proyecto abarca el desarrollo de un software de análisis de señales ECG con técnicas avanzadas de machine learning, pero se enfrenta a importantes limitaciones como la falta de validación experta y posibles restricciones en la disponibilidad de datos. A pesar de estos desafíos, el proyecto tiene como objetivo proporcionar al campo de la salud una herramienta útil para analizar preliminarmente señales electrocardiográficas y comparar métodos de machine learning.

Capítulo 2

Fundamentos Teóricos del Síndrome Metabólico

2.1. Generalidades

El síndrome metabólico (SM) es un grupo de trastornos que se presentan al mismo tiempo y aumentan el riesgo de enfermedad cardíaca, accidente cerebrovascular y diabetes tipo 2. Estos trastornos incluyen aumento de la presión arterial, niveles altos de azúcar en sangre, exceso de grasa corporal alrededor de la cintura y niveles anormales de colesterol o triglicéridos [9].

Tener solo uno de estos trastornos no significa que tengas síndrome metabólico. Pero sí significa que tienes un mayor riesgo de contraer una enfermedad grave. Y si desarrollas más de estos trastornos, tu riesgo de complicaciones, como la diabetes tipo 2 y las enfermedades cardíacas, aumenta aún más.

El síndrome metabólico es cada vez más frecuente, y hasta un tercio de los adultos estadounidenses lo tienen. Si tienes síndrome metabólico o cualquiera de sus componentes, los cambios radicales en el estilo de vida pueden demorar o incluso evitar la aparición de problemas de salud graves [9].

El SM en pacientes aumenta en cinco veces el riesgo de sufrir diabetes mellitus tipo 2 y en dos veces el riesgo de desarrollar una enfermedad cardiovascular (ECV) en los próximos 5 a 10 años comparados con individuos sin SM. Además, los pacientes con dicho síndrome tienen un riesgo de dos a cuatro veces de sufrir derrame cerebral y de tres a cuatro veces de sufrir infarto al miocardio.

La etiología de la ECV en pacientes con SM puede involucrar: enfermedad aterosclerótica coronaria, hipertensión arterial, hipertrofia del ventrículo izquierdo, disfunción diastólica, disfunción endotelial, enfermedad micro-vascular coronaria y disfunción autonómica. La patogénesis de la ECV en el SM es multifactorial y puede ser causada por uno o más factores asociados con esta condición, tales como la resistencia a la insulina, la diabetes o la inflamación crónica. Una característica común de la ECV en el SM y la resistencia a la insulina es la presencia de estrés oxidativo aumentado en el corazón [10].

2.2. Epidemiología

La presencia del SM se asocia con la edad, con la actividad física disminuida, con la dislipidemia, la hipertensión, con el tratamiento con antihiperglicémicos orales y con los niveles de HBA1c $\geq 7.0\,\%$. El riesgo es mayor en las mujeres, en los pacientes con niveles elevados de glucosa en ayunas y en los desórdenes endocrinos [11].

La emergencia de la obesidad y del SM en países en desarrollo se relaciona con un número de factores. La transición demográfica (hacia menor fertilidad, menor mortalidad y mayor expectativa de vida) y la transición epidemiológica (de una prevalencia por enfermedades infecciosas a un patrón de enfermedades relacionadas con el estilo de vida) han llevado a cambios significativos en la dieta y en la actividad física. Estos cambios causan efectos significativos sobre la composición corporal y el metabolismo, resultando frecuentemente en aumentos en el IMC, en la obesidad abdominal y en la dislipidemia [12].

La edad, la etnicidad y el sexo contribuyen a la susceptibilidad metabólica, en parte mediada por diferencias en la distribución del tejido adiposo y del tamaño del adipocito. Por ejemplo, los asiáticos del sur tienen un contenido de grasa corporal, una relación cintura / cadera, una relación grasa visceral / grasa subcutánea y un área de los adipocitos mayor que los caucásicos ajustados por edad, sexo e índice de masa corporal(IMC)[13][14].

2.3. ¿A quiénes y cómo evaluar el síndrome metabólico?

La evaluación del síndrome metabólico debiera realizarse a personas obesas, aquellos con diagnóstico de dislipidemia, intolerancia a la glucosa, hipertensión y diabetes. La importancia de la evaluación de SM en diabéticos tipo 2, radica en que hay evidencia de reducción de riesgo cardiovascular en ausencia de SM, además de ser los factores de riesgo para el desarrollo de síndrome metabólico los mismos que para desarrollo de enfermedad cardiovascular o diabetes [15][16]. Sugerimos entonces buscar SM en personas con factores

de riesgo de desarrollar diabetes, como los descritos por la American Diabetes Association (ADA), entre ellos, personas que no realicen o tengan escasa actividad física, con antecedente familiar de diabetes o enfermedad cardiovascular y mujeres con ovario poliquístico o madres con hijos macrosómicos (más de 4,1 kg), considerando riesgo bajo o moderado según corresponda por el número de factores presentes [17].

La evaluación del SM debe sustentarse en una buena historia clínica donde se evalúe los antecedentes de la persona y se realice un buen examen físico. La evaluación correcta de cada componente del SM nos llevará a buen diagnóstico y a detectar personas de riesgo alto que pudieran no estar siendo consideradas como tal, siendo tratadas por factores de riesgo individuales.

Las siguientes son sugerencias para una buena evaluación:

- Evaluación del perímetro abdominal: debe realizarse con el paciente en posición de pie al final de una espiración normal, con los brazos relajados a cada lado. La medida debe tomarse a la altura de la línea media axilar, en el punto imaginario que se encuentra entre la parte inferior de la última costilla y el punto más alto de la cresta ilíaca (principal punto de referencia) [18].

- Determinación de glicemia en ayunas: debe realizarse con por lo menos ocho horas previas de ayuno y en las primeras horas de la mañana, pues sabemos que fisiológicamente nuestro organismo tendrá una respuesta hepática compensatoria si no ingerimos alimentos y la medición no será exacta. De igual forma, fisiológicamente tendremos una concentración de glucosa elevada para nuestra referencia si no guardamos el ayuno respectivo, mostrando los resultados valores posprandiales, para los cuales las referencias aceptadas son diferentes [19].

- Determinación de triglicéridos y de colesterol HDL: también debe realizarse con por lo menos ocho horas previas de ayuno y en las primeras horas de la mañana. La concentración de triglicéridos puede variar según lo descrito para glucosa, no así el valor de HDL. Sin embargo, se recomienda que la medición de ambos sea en ayunas [19].

- Medición de la presión arterial: debe realizarse cuando la persona esté descansada y tranquila. No debe tomarse después del ejercicio o si la persona se siente estresada. Recordemos que no estamos diagnosticando hipertensión arterial. Por tanto, nuestro objetivo debe ser claro respecto al valor de referencia (130/85 mmHg) para el diagnóstico de SM. La medición podemos realizarla usando un monitor digital para presión arterial o un esfigmomanómetro y estetoscopio. La medición va a ser correcta en ambos casos si realizamos el procedimiento de manera adecuada. Podemos utilizar

la Recomendaciones de la American Heart Association para la toma de la presión arterial", publicado en la revista Circulation de febrero del año 2005 [19].

Todas las sugerencias descritas para el diagnóstico de SM son en ausencia de diagnóstico de diabetes, hipertensión arterial y dislipidemia, las cuales son condiciones definidas como componentes para el diagnóstico de SM.

2.4. Prueba oral de tolerancia a la glucosa

La Prueba de Tolerancia Oral a la Glucosa (POTG) es una prueba que se realiza por la mañana, se suministra una muestra de glucosa (75 g de glucosa), en ayunas, 8-14 horas después de la última comida en una persona después de haber tenido una buena noche de descanso y haber seguido una dieta común con un consumo normal de carbohidratos durante 3 días. Es necesario verificar si el paciente está tomando medicamentos que puedan aumentar los niveles de glucosa en sangre (como corticosteroides, diuréticos, tiazídicos y ciertos β-bloqueantes)[20].

Después de tomar una muestra de sangre venosa en ayunas, el paciente ingiere 250-300 ml de agua con 75 g de glucosa anhidra disuelta durante 5 minutos. Luego, el paciente debe permanecer sentado durante 120 minutos sin realizar esfuerzo físico, fumar o comer antes de que se le extraiga sangre. Si se mide la insulina al mismo tiempo, las mediciones pueden hacerse cada 60 minutos o incluso cada 30 minutos. Cambios en la respuesta a esta prueba son un signo distintivo del síndrome metabólico [20].

Capítulo 3

Fundamentos Teóricos de señales electrocardiográficas (ECG)

3.1. Generalidades

El ECG es un examen que evalúa las señales eléctricas responsables de la actividad del corazón. Midiendo la forma en que los impulsos eléctricos se desplazan por el músculo cardíaco durante su contracción y relajación. Durante un electrocardiograma, se colocan pequeñas almohadillas (electrodos) en la piel del pecho, los brazos y las piernas. También se conectan los electrodos a una máquina que convierte la actividad eléctrica en trazados sobre papel [21].

El ECG convencional muestra 12 derivaciones que representan la actividad eléctrica del corazón a través de las diferencias de potencial entre electrodos positivos y negativos ubicados en el cuerpo. Seis de estas derivaciones son verticales, usando las derivaciones frontales I, II y III, así como las derivaciones de los miembros aVR, aVL y aVF; mientras que 6 son horizontales, utilizando las derivaciones precordiales V1, V2, V3, V4, V5 y V6 [21]. El ECG de 12 derivaciones resulta fundamental para establecer muchos diagnósticos cardíacos, incluidos:

- Arritmias
- Dilatación de la aurícula
- Condiciones que predisponen a síncope o muerte súbita (síndrome de Brugada, síndrome de QT largo, Síndrome de Wolff-Parkinson-White)
- Isquemia miocárdica

3.2. Componentes tradicionales del ECG

Por convención, el trazado electrocardiográfico se divide en la onda P, el intervalo PR, el complejo QRS, el intervalo QT, el segmento ST, la onda T y la onda U como se puede observar en la figura 3.1.

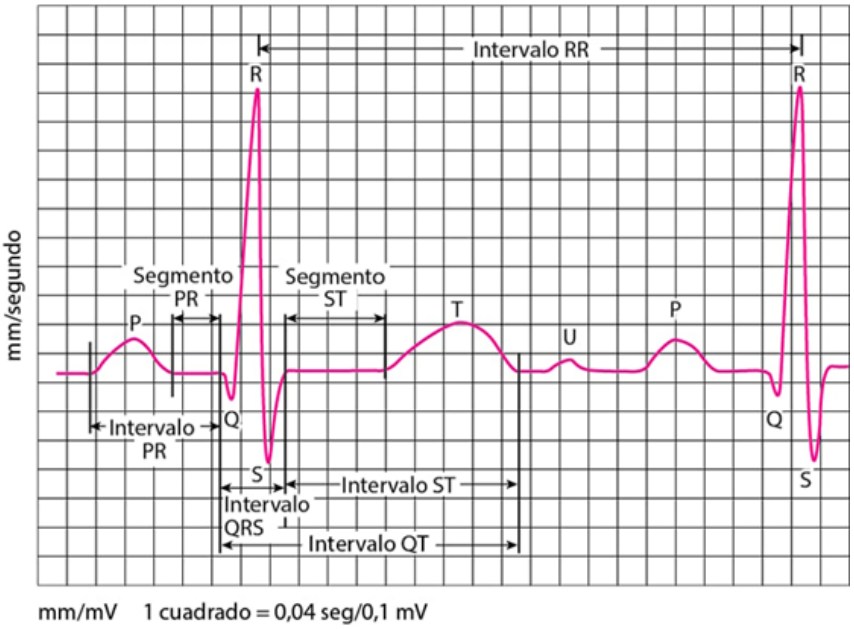


Figura 3.1: Componentes de una señal electrocardiográfica [22].

La activación de las aurículas está representada por la onda P. El intervalo PR muestra el lapso entre el comienzo de la despolarización auricular y la despolarización ventricular. Las ondas Q, R y S son los componentes del complejo QRS que refleja la despolarización de los ventrículos. El tiempo entre el comienzo de la despolarización ventricular y el final de su repolarización es medido por el intervalo QT. El intervalo RR es el lapso de tiempo entre dos complejos QRS seguidos. El indicador de la repolarización ventricular es la onda T, y el segmento ST junto con la onda T (ST-T) también señalan este proceso. El período posterior a la despolarización (relajación) de los ventrículos probablemente se asocie con la onda U [22].

3.2.1. Onda P

La despolarización de las aurículas es representada por la onda P del electrocardiograma. En la mayoría de las derivaciones, esta onda sube, salvo en el caso de la derivación aVR. En II y V1, la forma bifásica puede estar presente; el primer componente refleja

la actividad de la aurícula derecha, mientras que el segundo muestra la actividad de la aurícula izquierda.

Si las aurículas aumentan de tamaño debido a hipertrofia o dilatación, también es posible que la amplitud de uno o ambos componentes de la onda P aumenta. Cuando la aurícula derecha se hipertrofia, la onda P supera los 2 mm en las derivaciones II, III y aVF, conocida como "P pulmonar". Además, la hipertrofia auricular izquierda se muestra como una onda P ancha con dos picos en la derivación II, conocida como "P mitral". El eje de la onda P generalmente está inclinado entre 0° y 75° con respecto a los ventrículos [22].

3.2.2. Intervalo PR

El intervalo PR es el período entre el comienzo de la despolarización auricular y la despolarización ventricular. En condiciones normales, dura entre 0,10 y 0,20 segundos y su prolongación define el bloqueo auriculoventricular de primer grado [22].

3.2.3. Complejo QRS

La despolarización ventricular es representada por el complejo QRS. En todas las derivaciones, excepto en V1-V3, una duración normal de la onda Q es inferior a 0,05 segundos y se considera anormal si indica un infarto de miocardio actual o pasado. La primera deflexión ascendente es la onda R; no se han establecido criterios absolutos para su altura o tamaño normal, aunque las ondas R más bajas pueden ser causadas por hipertrofia ventricular. Se llama R' a una segunda deflexión ascendente en el complejo QRS. Cuando no hay onda Q, la onda S es la primera deflexión descendente; si está presente la onda Q, entonces es la segunda.

El complejo QRS puede estar compuesto solo por la onda R, las ondas QS (sin R), las ondas QR (sin S), las ondas RS (sin Q) o por RSR' dependiendo de la derivación del ECG, el vector y la presencia de enfermedades cardíacas. El complejo QRS tiene una duración de $0,07$ a $0,10$ segundos en condiciones normales. Un intervalo de $0,10$ a $0,11$ segundos representa un bloqueo parcial en una rama del haz de His o un retraso inespecífico en la conducción intraventricular, definido por la morfología del complejo QRS. Se considera bloqueo de rama completo o retraso de la conducción intraventricular si el intervalo es $\geq$ 0,12 seg.

El eje normal del complejo QRS oscila entre 90° y $-30°$. Un eje de entre $-30°$ e $-90°$ indica una desviación a la izquierda y se identifica en el bloqueo del fascículo anterior

izquierdo ($-60°$) y en el infarto de miocardio inferior. Un eje de entre 90° e 180° implica una desviación a la derecha y se presenta en cualquier trastorno que aumente las presiones pulmonares y produzca hipertrofia ventricular derecha (pulmonar, embolia pulmonar aguda, hipertensión pulmonar), así como también en el bloqueo de la rama derecha o posterior izquierda del fascículo de His [22].

3.2.4. Intervalo QTc

El intervalo QT es el período entre el comienzo de la despolarización ventricular y el final de la repolarización ventricular. El intervalo QT debe corregirse en función de la frecuencia cardíaca a través de la siguiente fórmula:

$$QT_c = \frac{QT}{\sqrt{RR}}$$

Donde QTc es el intervalo QT corregido y el intervalo RR es el período entre dos complejos QRS. Todos los intervalos se registran en segundos [22].

3.2.5. Segmento ST

El segmento ST del electrocardiograma muestra la despolarización total del músculo ventricular. Bajo condiciones normales, este segmento es horizontal y se encuentra a lo largo de la línea basal de los intervalos PR o TP, o ligeramente debajo de esta [22].

3.2.6. Onda T

La onda T refleja la repolarización ventricular. En general, adopta la misma dirección que el complejo QRS (concordancia). La polaridad opuesta (discordancia) puede indicar un infarto actual o pasado. La onda T suele ser regular y redondeada, pero puede ser de baja amplitud en pacientes con hipopotasemia e hipomagnesemia o alta y "picuda.[en] individuos con hiperpotasemia, hipocalcemia e hipertrofia ventricular izquierda [22].

Capítulo 4

Fundamentos Teóricos de filtrado de señales

4.1. Generalidades

Los filtros son dispositivos que modifican de algún modo determinado una señal que pase por medio de él. Se conocen diferentes clasificaciones de los filtros, como lo son: filtros eléctricos, filtros mecánicos, filtros acústicos, filtros ópticos, entre otros. Asimismo, se tienen clasificaciones orientadas a las respuestas otorgadas por dichos dispositivos, a su comportamiento y su modelación matemática, estos son los filtros lineales y los filtros no lineales. La clasificación que más compete al proyecto es la de los filtros analógicos y los filtros digitales, el primero de estos es para aquellos donde la señal puede tomar cualquier valor durante un intervalo determinado; los segundos son aquellos donde solo la señal toma valores discretos [23]. Una representación gráfica, de lo anteriormente descrito, se puede observar en la figura 4.1.

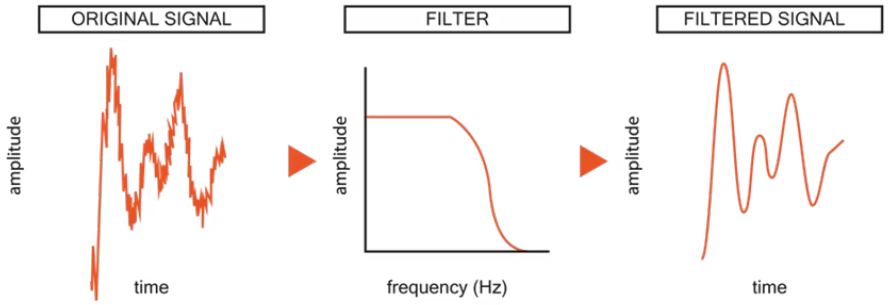


Figura 4.1: Filtrado de señales [24].

4.2. Filtros digitales

Los filtros digitales son LTI, es decir, sistemas lineales e invariantes en el tiempo, los cuales cambian el comportamiento de la frecuencia de una señal presente a la entrada, todo esto con base en la respuesta que tenga en frecuencia, también a esto se le conoce como la función de transferencia. Obteniendo a su salida una señal que multiplica las señales de entrada y la señal de respuesta en frecuencia. Estos filtros se dividen en dos tipos [25]:

- *Filtro de Respuesta Finita al Impulso (FIR):* Este se caracteriza por no ser recursivo, posee fase lineal, siempre cuentan con estabilidad, no presentan problemas de desbordamiento, entre otras.
- *Filtro de Respuesta Infinita al Impulso (IIR):* Su característica principal es la retroalimentación presente desde la señal de salida. Su salida no solo es función de la entrada y los elementos anteriores, sino también de las salidas anteriores.

4.3. Filtros Ideales

Los filtros ideales son selectores de frecuencia que permiten el paso sin ningún tipo de distorsión de las componentes presentes en el espectro, las cuales están comprendidas en la o las bandas de paso, eliminando por completo las componentes que no se encuentran dentro de ellas [23]. Se presentan varios tipos de filtros:

- Filtro pasa bajos.
- Filtro pasa altos.
- Filtro pasabanda.

4.4. Tipo de filtros

4.4.1. Filtros pasabajos

Son aquellos que permiten el paso de las frecuencias inferiores a la frecuencia de corte ω_c, e impide el paso a aquellas frecuencias que se encuentran por encima de la frecuencia de corte [23].

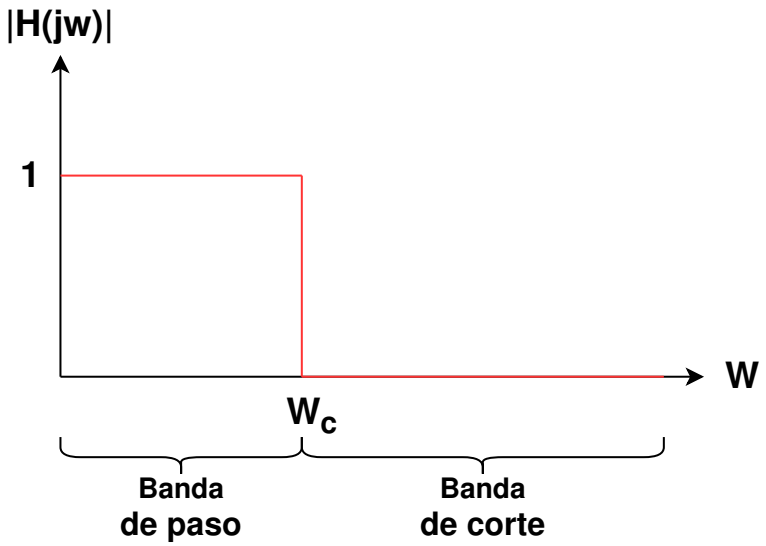


Figura 4.2: Respuesta filtro pasa bajos.

4.4.2. Filtros pasaaltos

Son aquellos que permiten el paso de las frecuencias superiores a la frecuencia de corte ω_c, e impiden el paso a aquellas frecuencias que se encuentran por debajo de la frecuencia de corte [23].

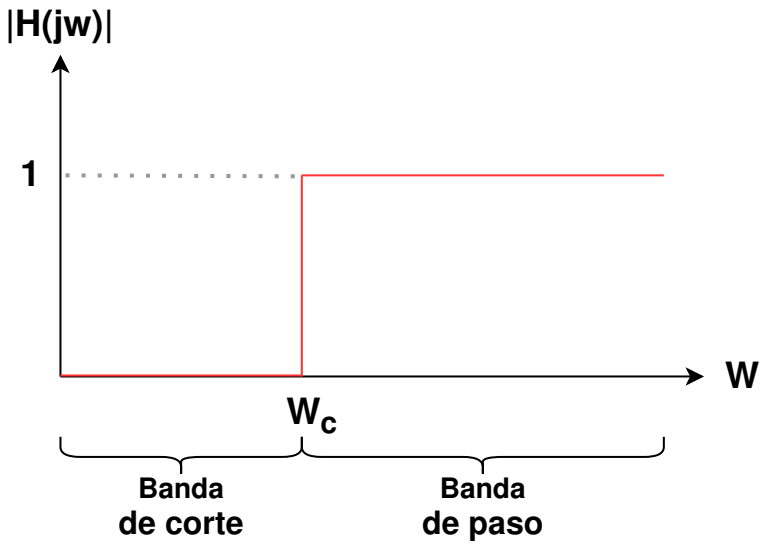


Figura 4.3: Respuesta filtro pasaaltos.

4.4.3. Filtros pasabanda

Son aquellos que permiten el paso de toda aquella frecuencia que se encuentre entre la frecuencia inferior de corte y la frecuencia superior de corte (ω_1 y ω_2 respectivamente) [23].

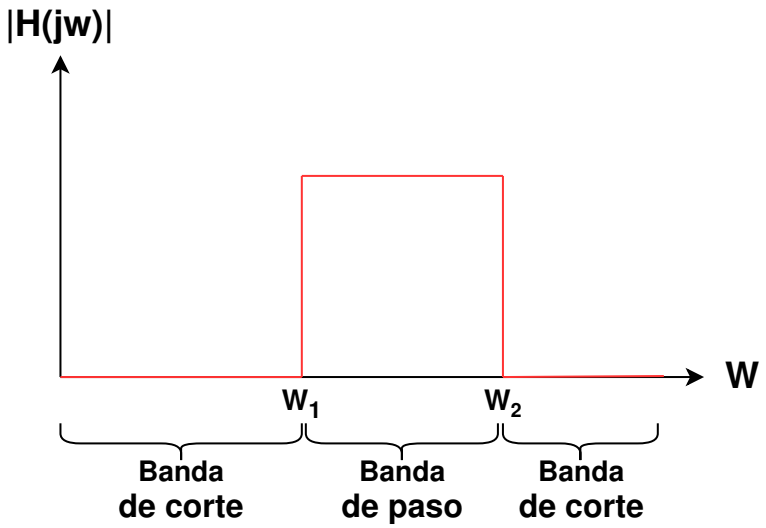


Figura 4.4: Respuesta filtro pasabanda.

4.4.4. Filtro Butterworth

"Los filtros paso bajo de Butterworth son filtros de solo polos caracterizados por el módulo al cuadrado de la respuesta en frecuencia"[26, pág. 639]. Esto se puede observar en la siguiente ecuación:

$$|H(\Omega)|^2 = \frac{1}{1 + (\frac{\Omega}{\Omega_c})^{2N}} = \frac{1}{1 + \varepsilon^2(\frac{\Omega}{\Omega_p})^{2N}} \tag{4.1}$$

Donde N es el orden del filtro, Ω_c es su frecuencia de corte, Ω_p es la frecuencia de corte de la banda de paso y $\frac{1}{1+\varepsilon^2}$ es el valor en la frecuencia de corte en la banda [26].

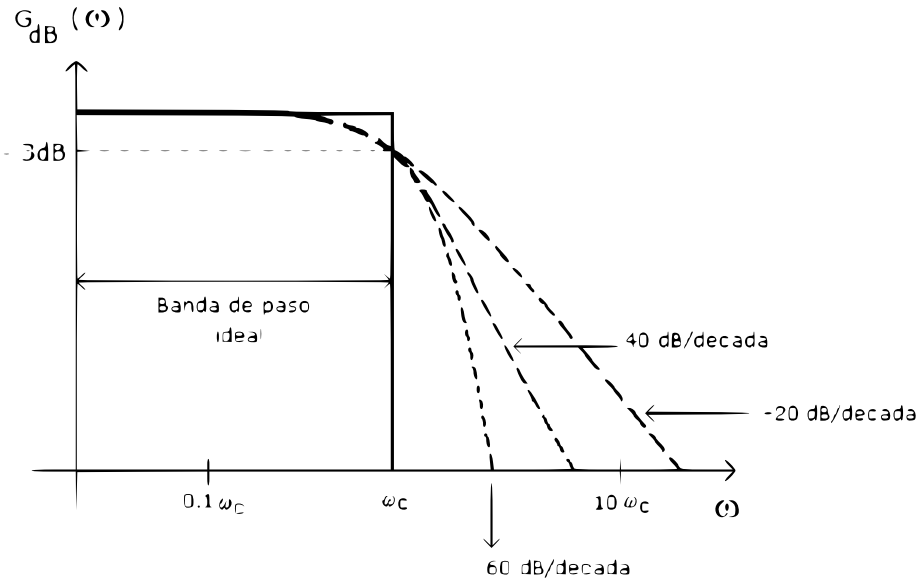


Figura 4.5: Diagrama de Bode del filtro Butterworth [27].

Capítulo 5

Fundamentos Teóricos de Machine Learning

5.1. Generalidades

La tecnología de Machine Learning (ML) ha permitido la creación y generación de nuevos modelos más rápidos y automáticos, los cuales tienen la capacidad de analizar mayor cantidad de datos, presentando estos cierto nivel de complejidad; produciendo así, resultados de manera más veloz y precisa. Estos procesos generan una automatización de las tareas o acciones a realizar por parte de la máquina, en donde esta logra aprender de manera autónoma y según que tipo de ML se implemente, no depende de una constante alimentación de datos. Una de las definiciones más acertadas acerca del ML y que engloba en su sintaxis la esencia del proyecto en cuestión es: "El Machine Learning es una técnica asociada a la detección automática de patrones relevantes dentro de un conjunto de datos." [28, pág. 27]. Alejándose de las definiciones netamente relacionadas con el proyecto, se encuentran descripciones las cuales afrontan de forma generalizada el concepto de ML, "Machine Learning es la ciencia que permite que las computadoras aprendan y actúen como lo hacen los humanos, mejorando su aprendizaje con datos e información en forma de observaciones e interacciones con el mundo real" [29, pág. 3]. Uno de los fundamentos primordiales para lograr la correcta implementación y ejecución del ML, es el aprendizaje.

5.2. El aprendizaje de un sistema de Machine Learning

La forma por medio de la cual un sistema de ML realiza su aprendizaje es la generación de un modelo que cuenta con ciertos parámetros específicos, los cuales se modifican de distintas maneras con el fin que estos llegasen a estar en una función de los diferentes datos a manipular. Estos datos presentan tanto variables dependientes como variables independientes, las variables independientes son las encargadas de generar un modelo que es capaz de predecir, de la mejor manera como sea posible, las variables dependientes; todo este proceso de instrucciones de autoaprendizaje o predicción, se lleva a cabo por medio de un algoritmo.

El ML se diferencia de la programación clásica, debido a que no presenta un conjunto de reglas previamente diseñadas por un programador, el cual busca obtener una respuesta determinada. Por el contrario, el ML, elabora un conjunto de reglas más efectivas, las cuales se encargan de asociar los datos recibidos a su entrada con las respuestas que se esperan adquirir a través del aprendizaje [28].

5.2.1. Tipos de modelos de Machine Learning

Los sistemas de Machine Learning tienen diferentes maneras de clasificarse, esto se hace mediante las formas de aprendizaje que existen para los ML, los cuales, por lo general, se catalogan en cuatro [30]:

- *Aprendizaje supervisado:* Es aquel modelo en donde el algoritmo aprende de los datos y respuestas previamente suministrados, llegando así a predecir posteriormente la respuesta correcta al momento de que se presentase una nueva muestra.
- *Aprendizaje no supervisado:* El algoritmo desarrolla la habilidad de aprender por sí solo el comportamiento del sistema en cuestión, a partir de ejemplos simples, es decir, el mismo algoritmo se encarga de descubrir el método por el cual llegará a la respuesta correcta.
- *Aprendizaje reforzado:* Es un aprendizaje donde se carece de datos, valores, conjuntos, clases, entre otros parámetros; previamente dados. El algoritmo determina cómo funciona y cómo debe ser su comportamiento, a través de la prueba y error.
- *Aprendizaje profundo:* Está basado en aprendizaje por varios niveles, los cuales se encargan de representar y abstraer con el fin de modelar relaciones complejas entre los conjuntos de datos [28].

Aprendizaje Supervisado

Profundizando más en el modelo de aprendizaje supervisado, este "se utiliza para predecir una variable de salida (objetivo) a partir de un conjunto de variables de entrada y un conjunto de datos de entrenamiento previamente etiquetados." [31, pág. 8] El modelo de aprendizaje supervisado presenta dos problemas comunes, los cuales busca resolver: los problemas de regresión, siendo estos los encargados de predecir una cantidad continua de datos o elementos; y los problemas de clasificación, quienes son los encargados de predecir una etiqueta [28].

Los algoritmos más utilizados son:

- La regresión lineal.
- La regresión polinomial.

Los algoritmos de clasificación más utilizados son:

- La regresión logística.
- Support Vector Machine (SVM).
- K-Nearest Forest.
- Random Forest (RF).

Support Vector Machine (SVM)

Las Máquinas de Vectores de Soporte, tienen como objetivo definir un hiperplano en un espacio de N dimensiones, en donde este es el límite de decisión, permitiendo así la clasificación de los datos [28]. Brindando un conjunto de datos de entrenamiento, donde cada uno de estos tienen unas características y etiquetas pertenecientes a una clase determinada, con la finalidad de encontrar el hiperplano de separación que maximice el margen entre los conjuntos de datos de cada clase [31]. "El margen se define como la distancia perpendicular más cercana entre el hiperplano y las muestras de entrenamiento." [31, pág. 12].

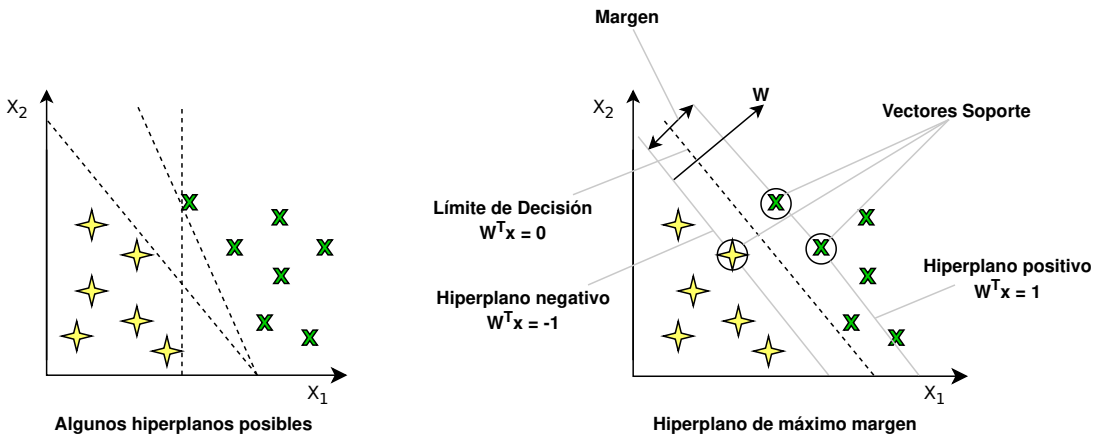


Figura 5.1: Representación gráfica de una Máquina de Vectores de Soporte.

En conjuntos de datos que poseen dos dimensiones, se presentan dos tipos de clasificadores [30]:

- *Problemas de clasificación linealmente separables:* En este caso se organizan los datos dependiendo de su ubicación en el hiperplano [18]. En donde, si la salida es mayor a 1 se le asigna una clase a ese dato, por el contrario, si la salida es menor a 1 se le asigna otra clase diferente a este nuevo dato. Dando como resultado que el rango que actúa como margen es -1,1 [28].

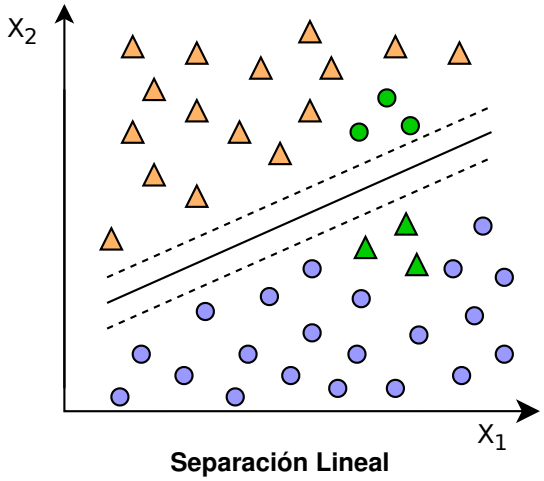


Figura 5.2: Separación lineal de los datos en SVM.

- *Problemas de clasificación no linealmente separables:* Este caso se presenta en la gran mayoría de situaciones, para realizar dicha separación, se hace uso de herramientas matemáticas, las cuales permiten realizar el mapeado de los datos en un espacio dimensionalmente mayor, en donde allí se logran separar linealmente. Dichas herramientas reciben el nombre de funciones de Kernel, las cuales se encargan de calcular "el producto escalar entre las muestras en el espacio de características ampliado sin necesidad de realizar la transformación explícita." [31, pág. 13].

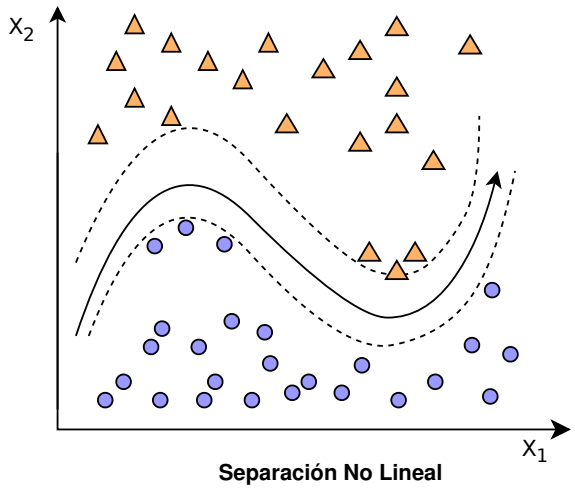


Figura 5.3: Separación no lineal de los datos en SVM.

Se pueden llegar a presentar pequeños errores al momento de obtener la separación óptima del hiperplano, en donde se llega a tener una serie de muestras o conjunto de datos atípicos o ruido en los datos. Es así, como existen una serie de datos llamados vectores de soporte, los cuales son de suma importancia para determinar la ubicación y orientación del plano. Además, permiten realizar predicciones más precisas de nuevos conjuntos de datos [31].

Random Forest (RF)

Los árboles de decisión, son algoritmos de aprendizaje automático que llevan este nombre debido a la estructura que describen, este toma decisiones basadas en reglas lógicas. Este está constituido por nodos que representan características, y arcos que representan las decisiones fundamentadas en las características de los nodos. La característica de mayor importancia se denomina nodo raíz, este tiene el mayor poder de discriminación. Además de este nodo raíz se tienen otros dos nodos, los cuales son: los nodos internos, encargados de representar las pruebas en las características, y los nodos hoja, encargados de representar las clases o valores de salida.

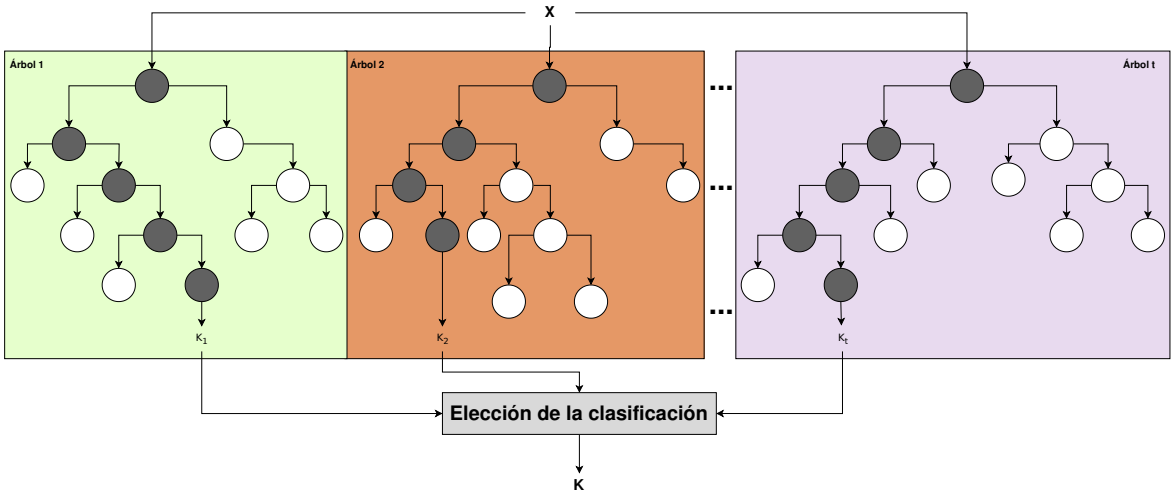


Figura 5.4: Estructura de un árbol de decisión (RF).

Las decisiones en este tipo de algoritmo se toman siguiendo una ruta específica, que va desde el nodo raíz hasta el nodo hoja, es decir, va desde el centro y la característica principal hasta el punto de salida de las variables o valores; durante el camino recorrido debe cumplir con cada una de las reglas de decisión de los nodos internos. Estas reglas de decisión o decisiones, se pueden asemejar a elementos de la programación clásica, como lo es la anidación de funciones if, en donde si un atributo o variable cumple con cierta característica se debe dirigir a cierta parte del árbol, de lo contrario irá por la parte opuesta.

El árbol se construye dividiendo el conjunto de datos de entrenamiento en subconjuntos de menor tamaño que se están sustentados en los atributos, buscando la mayor homogeneidad posible en los términos de la clase. Se debe realizar esta división hasta obtener un valor máximo de profundidad del árbol, todo por medio de la recursividad [31].

5.3. Aplicaciones del aprendizaje supervisado

Se han realizado numerosas aplicaciones del aprendizaje supervisado en diferentes áreas, unas de estas son [31]:

- Reconocimiento de voz.
- Reconocimiento de imágenes.
- Clasificación de texto.

- Detección de fraude.
- Diagnósticos médicos.

5.4. Preprocesamiento de datos

Este es el primer paso del análisis de datos, el cual permite procesar datos a un formato que sea comprensible y se pueda utilizar para el análisis. Esto con el fin de garantizar que los datos a utilizar estén preparados y listos para las etapas que se realizarán posteriormente. La precisión y/o fiabilidad del preprocesamiento de datos genera la confiabilidad y exactitud del análisis de datos. Este proceso determina el fácil y correcto uso e interpretación de los datos, sirviendo como punto de partida para los modelos de aprendizaje automático e inteligencia artificial [32].

Este preprocesamiento de datos consta de varias etapas: [32]

- **Perfil de los datos:** Este primer paso permite realizar la examinación de los datos por medio de métodos estadísticos con el fin de comprender su estructura, contenido y calidad, con el objetivo de revelar patrones, anomalías y correlaciones.
- **Limpieza de datos:** En este segundo paso se detectan y corrigen los registros de los datos corruptos o inexactos, valores atípicos, duplicados y valores que falten, esto con el fin de garantizar la precisión del conjunto de datos.

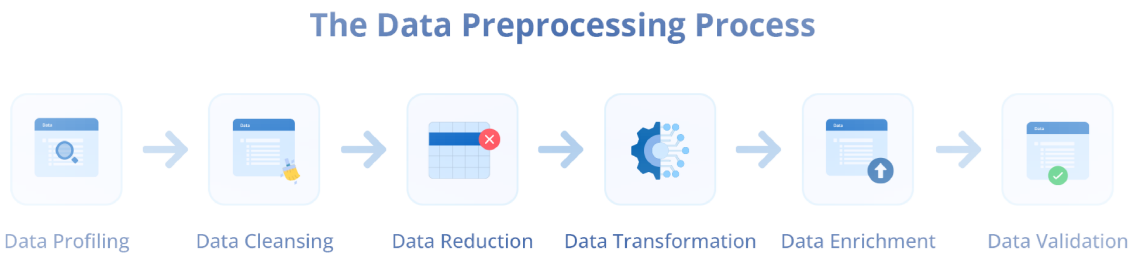


Figura 5.5: Etapas del preprocesamiento de datos [32].

- **Reducción de datos:** El tercer paso busca reducir la gran cantidad de datos, con el fin de producir resultados analíticos iguales o similares. Se hace uso de técnicas

como la reducción de dimensionalidad, agrupación, histogramas y el análisis de componentes principales, que permiten realizar una simplificación de los datos sin perder patrones y tendencias informativas.

- **Transformación de datos:** Cambia el conjunto de datos para las necesidades que se tengan posteriormente. También se puede utilizar para crear un nuevo conjunto de atributos, los cuales permiten obtener diferentes funciones para extraer ciertas características determinadas.
- **Enriquecimiento de datos:** La mejora de los datos proporciona mayor profundidad y contexto, esto se realiza por medio de fuentes adicionales o atributos derivados.
- **Validación de datos:** Se realiza como paso previo al análisis de los datos, con la finalidad de garantizar que se cumpla con los criterios específicos necesarios. Por medio de este paso, se puede corroborar la precisión de los datos, que estén completos y su confiabilidad.

Existen diferentes técnicas de preprocesamiento de datos como lo son [32]:

- Imputación de datos.
- Reducción de datos ruidosos.
- Identificación y eliminación de datos duplicados.
- Ingeniería de características.
- Escalado o normalización de características.
- Reducción de dimensionalidad.
- Discretización.
- Codificación de funciones.

5.4.1. Conceptos estadísticos para el preprocesamiento de datos

Como se mencionó anteriormente en la sección de preprocesamiento de datos, el primer paso a realizar consta de una perfilación de datos, donde se realiza una examinación de estos por medio de métodos estadísticos. Algunos de estos métodos o conceptos pueden ser la moda, la media, la mediana, el error típico, la desviación estándar, el rango, la distribución normal, entre otros... por medio de estos se puede comprender la estructura, conocer la calidad de los datos y observar que patrones, características y correlaciones presentan

dicho conjunto de datos.

Existen métodos o pruebas los cuales permiten encontrar o determinar características específicas del conjunto de datos a estudiar. Estos pueden ser:

- Prueba de Kolmogórov-Smirnov.
- Prueba Mann-Whitney.

Prueba de Kolmogórov-Smirnov

Es un método utilizado comúnmente para realizar la prueba de normalidad de los datos muestrales, se hace uso de esta para proceso no lineales e interactivos, debido a que estos presentan, por lo general, distribuciones no gaussianas. Además, se debe contar con el prerrequisito de normalidad para implementar algunas pruebas de este mismo concepto. Por medio de esta prueba en mención, se busca comprobar la hipótesis de normalidad de una población determinada, esto se ve representado en la siguiente ecuación [33]:

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n}\begin{cases} 1 : \text{sí } y_i \leq x_i \\ 0 : alternativa \end{cases} \tag{5.1}$$

Para dos colas el estadístico está determinado por las siguientes ecuaciones:

$$D_n^+ = max(F_n(x) - F(x)) \tag{5.2}$$

$$D_n^- = max(F(x) - F_n(x)) \tag{5.3}$$

Donde $F(x)$ es la distribución presentada por la hipótesis.

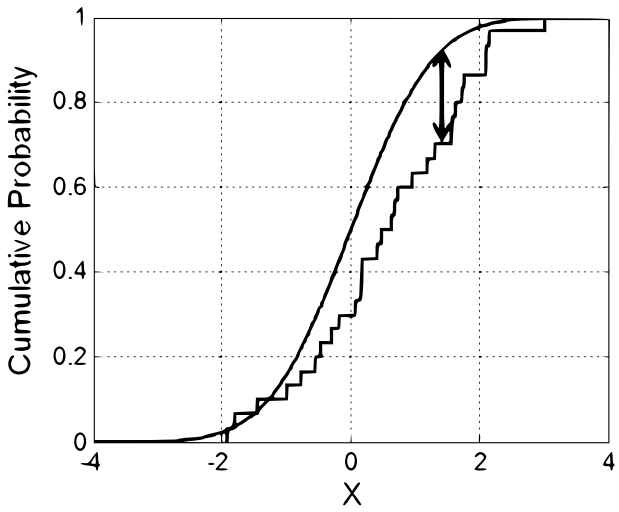


Figura 5.6: Ejemplo prueba de Kolmogórov-Smirnov [34].

Estas ecuaciones 5.2 y 5.3, tienen una explicación "La distribución del estadístico de Kolmogórov-Smirnov es independiente de la distribución poblacional especificada en la hipótesis nula y los valores críticos de este estadístico están tabulados" [33, pág. 89]. Además, la prueba de Kolmogórov-Smirnov, cuenta con una serie de valores que se encuentran estipulados en sus tablas, las cuales son válidas únicamente si se prueba que el conjunto de observaciones presente viene de una distribución continua completamente especificada.

Prueba de Mann-Whitney

Esta prueba se emplea para realizar la comparación entre dos muestras independientes, bien sea con variables cuantitativas o variables cualitativas ordinales [35]. Por medio de esta se obtienen diferentes conclusiones sobre los datos en función de las suposiciones que se hagan sobre la distribución de los mismos [36].

Los datos a utilizar para realizar la prueba de Mann-Whitney deben tener una serie de características para poder ser analizados [36]:

- La variable dependiente debe medirse a nivel ordinal o continuo.
- La variable independiente debe consistir en dos grupos.
- Debe haber independencia de las observaciones de cada grupo.
- Las dos variables se distribuyen de forma anormal.

Capítulo 6

Estado del arte

El síndrome metabólico (SM) son un conjunto de factores los cuales pueden ser causantes de diferentes enfermedades, dentro de estas condiciones se encuentra la acumulación de grasa o tejido adiposo en áreas específicas como pueden ser la zona baja del abdomen y las paredes del corazón. Se estudió la posibilidad de que la aglomeración de grasa, en las zonas ya mencionadas, podría generar alteraciones en las señales eléctricas que controlan el ritmo cardiaco en pacientes con SM. De acuerdo con [37] comprobaron la relación entre estos, debido a que los resultados de los esquemas ECG fueron significativamente mayores en los pacientes con SM que en aquellos que no la padecían, llegando a este hallazgo por medio de la obtención del valor de la onda P, la cual se encontraba en rangos inferiores a 0.05; el nivel elevado de colesterol unido a lipoproteínas de alta densidad se asoció de forma negativa a los esquemas ECG. De igual forma, en [38] llevaron a cabo un estudio de las 12 derivaciones de los electrocardiogramas, en adultos mayores, los resultados indican una asociación significativa entre el síndrome metabólico y las anormalidades electrocardiográficas en dicha población. En [39] encontraron anomalías en las señales ECG, específicamente en la onda Q patológica y la hipertrofia ventricular izquierda, las cuales son indicadores de la existencia del SM en los pacientes. Además, se plantea la importancia del uso temprano de ECG para la detección prematura del SM y otras enfermedades relacionadas.

Después de confirmar la existencia de una relación entre las señales electrocardiográficas y el síndrome metabólico, se procede a investigar las variaciones presentes en dichas señales. En este sentido, se realiza un estudio enfocado en las ondas e intervalos que componen la señal y en [40] evidenció que dichos parámetros del electrocardiograma (ECG) mostraron diferencias significativas entre los participantes con y sin síndrome metabólico; la duración de la onda P, la duración del complejo QRS y el intervalo QT corregido fueron mayores [0.07 SD (0.05, 0.10), 0.04 SD (0.01, 0.06) y 0.05 SD (0.02, 0.08), respectivamente].

También se observó que los ejes de las ondas P, T y QRS fueron más bajos [-0.10 SD (-0.12, -0.07), -0.07 SD (-0.10, -0.05) y -0.19 SD (-0.22, -0.16), respectivamente], y el porcentaje de pequeñas ondas Q aumentó por cada componente adicional del síndrome metabólico. En complemento a estos estudios, [41] presenta una investigación donde propone de qué forma se ven afectados los intervalos que componen la señal, observaron que los individuos con síndrome metabólico presentaron intervalos QTc (corregido) y JTc (intervalo JT corregido) más prolongados en comparación con aquellos sin síndrome metabólico; además, se identificó una relación directa entre el número de parámetros metabólicos anormales y la longitud de los intervalos QTc y JTc; a medida que aumentaba el número de parámetros metabólicos anormales, estos intervalos también se prolongaba. Estos resultados refuerzan la importancia de entender la conexión entre el síndrome metabólico y las anomalías electrocardiográficas, proporcionando una base sólida para la investigación.

Actualmente, la prueba para determinar el síndrome metabólico es una prueba oral de tolerancia a la glucosa (POTG) la cual consta de 5 etapas, estas 5 etapas consisten en realizar una glucometría en ayunas y luego proporcionar una muestra de glucosa al paciente, proceso que se repite cada 30 minutos por dos horas. Al intentar aplicar el análisis de las señales ECG a la detección del síndrome metabólico se debe estudiar si estas señales reaccionan de manera diferente en sujetos de prueba sanos y sujetos con el síndrome metabólico. En [42] realizaron esta prueba en tres conjuntos, los cuales están compuestos por: 15 sujetos con síndrome metabólico, 15 sujetos deportista y 10 sujetos sedentarios, en este estudio, se comparó la variabilidad de la frecuencia cardiaca durante los primeros 15 min de cada una de las etapas de POTG, y gracias a esto se pudo concluir que los sujetos con síndrome metabólico presentaron reducción en todos los indicadores de la frecuencia cardiaca en comparación con los otros dos grupos. Lo que permite por medio de las variaciones de las señales ECG clasificar si un sujeto padece o no de estas complicaciones.

En [43] se propone el uso del método 1-D convolution el cual se usa para el reconocimiento de patrones de datos, esto con el fin de reducir el consumo de tiempo, ya que el método propuesto no requiere extracción manual de características de la señal ECG, debido a que las señales pasan directamente al modelo de aprendizaje profundo para predecir el SM. Para esto hacen uso de una base de datos proporcionada por [44], en la cual se encuentran las 12 derivaciones de la señal en 3 diferentes grupos, uno con SM, otro de deportistas y el último con sujetos sedentarios. Al final de este estudio se obtuvo una precisión del 88.32 %.

Capítulo 7

Desarrollo de Ingeniería

En este capítulo se detalla el desarrollo del trabajo a través de cinco secciones: en la primera, se presenta el análisis estadístico del conjunto de datos utilizado, con el propósito de caracterizar a los pacientes en términos de variables clínicas. Esto permite proporcionar un contexto claro sobre las personas a quienes se aplica el estudio y verificar que las características iniciales de los sujetos sean comparables. En la segunda sección, se describe el procesamiento de las señales ECG de los sujetos, cuyo objetivo es eliminar el ruido generado por los equipos de medición y resaltar algunos de los picos característicos de la señal ECG. La tercera sección aborda la detección de las series de tiempo, que proporcionan información detallada para identificar patologías de manera adecuada. En la cuarta sección, se calculan los parámetros en los dominios del tiempo, la frecuencia y no lineales a partir de las series obtenidas en la sección anterior. Por último, en la quinta sección, se aplican dos métodos de machine learning para el entrenamiento de los modelos y la realización de pruebas destinadas a sus predicciones. Además, se lleva a cabo una comparación entre los modelos empleados para evaluar cuál de ellos ofrece una mayor precisión.

7.1. Análisis estadístico del conjunto de datos

Como se ha mencionado en la sección 5.4, en la etapa inicial del preprocesamiento de datos "*perfil de los datos*", se realiza una examinación del conjunto de datos presentes por medio de métodos estadísticos. Es por esto que como primer paso, para iniciar el desarrollo del proyecto se lleva a cabo un análisis por medio de las herramientas estadísticas como *la media, la moda, la mediana, el error típico, distribución normal, entre otras...*

Estos procesos se aplican al conjunto de datos de los dos grupos de personas a las cuales se le realizaron los diferentes estudios tradicionales para determinar su condición respecto al síndrome metabólico. El primer conjunto de datos pertenece a las personas del grupo de

control, las cuales no presentan la enfermedad pero tienen condiciones que son un signo de alarma y deben estar en constate análisis para evitar complicaciones mayores en su salud; el segundo conjunto de datos pertenece a las personas del grupo que poseen el síndrome metabólico, las cuales ya han sido diagnosticadas previamente con dicha enfermedad. La decisión de extraer los datos de estos grupos, se debe a que por medio de estos se pueden realizar comparación en cada uno de los datos del proceso de diagnóstico, en donde al momento de entrenar a la ML, esta podrá diferenciar distintos aspectos y factores determinantes los cuales arrojen como resultado el estado y condición del paciente respecto al SM.

Estos dos grupos de estudio poseen datos correspondientes a los siguientes items:

- Altura.
- Peso.
- Área bajo la curva.
- IMC.
- Presión arterial sistólica.
- Presión arterial diastólica.
- Triglicéridos.
- Colesterol.
- LDL.
- HDL.
- VLDL.
- Circunferencia abdominal.
- ALT.
- Ácido úrico.
- Urea UV.
- HOMA.
- Glucosa en diferentes tiempos entre (0 y 120) minutos.

- Insulina en diferentes tiempos entre (0 y 120) minutos.

Algunos de estos items presentados, poseen un rango de normalidad para los cuales si la persona se encuentra fuera de dichos límites, no representan ningún signo de alarma y son indicadores de que dicha persona no padece o está cercana a padecer el SM. Estos items y rangos son:

- *Índice de masa corporal (IMC):* Entre (18.5 y 24.9) $\frac{kg}{m^2}$ [45].
- *Presión arterial sistólica:* Entre 90 y 120 [46].
- *Presión arterial diastólica:* Entre 60 y 80 [46].
- *Triglicéridos:* Menos de 150 $\frac{mg}{dL}$ [47].
- *Colesterol total:* Menos de 200 $\frac{mg}{dL}$ [48].
- *Colesterol LDL:* Menos de 100 $\frac{mg}{dL}$ [48].
- *Colesterol HDL:* Mayor a 60 $\frac{mg}{dL}$ [48].
- *Colesterol VLDL:* Entre (2 y 30) $\frac{mg}{dL}$ [49].
- *Circunferencia abdominal:* Máximo 88 centímetros en mujeres y máximo 102 centímetros en hombres [50].
- *ALT:* Entre (4 y 36) $\frac{U}{L}$ [51].
- *Ácido úrico:* Entre (3.5 y 7.2) $\frac{mg}{dL}$ [52].
- *Urea UV:* Entre (6 y 24) $\frac{mg}{dL}$ [53].
- *HOMA:* Valores por debajo de 2.7 [54].
- *Glucosa:* Entre (90 y 130) $\frac{mg}{dL}$ [55].
- *Insulina:* Entre (60 y 100) $\frac{mg}{dL}$ durante la noche y entre comida. Entre (60 y 140) $\frac{mg}{dL}$ después de las comidas [56].

Téngase en cuenta que cada uno de estos valores son medidas para personas adultas.

Como se mencionó anteriormente, a cada uno de estos items, se les realizó el análisis estadístico pertinente por medio de diferentes herramientas estadísticas, como se puede ver a continuación en la siguiente figura 7.1:

Grupo de SM

Análisis Presión Arterial Sistólica	
Media	135.13333
Error típico	1.8332035
Mediana	137
Moda	138
Desviación Estandar	7.0999665
Varianza de la muestra	50.409524
Curtosis	1.3547287
Coeficiente de simetría	-0.763675
Rango	29
Mínimo	118
Máximo	147
Suma	2027
Cuenta	15

Grupo de Control

Análisis presión arterial sistólica	
Media	117.6
Error típico	2.9181425
Mediana	116
Moda	110
Desviación Estandar	9.2279768
Varianza de la muestra	85.155556
Curtosis	1.6284929
Coeficiente de simetría	1.3899821
Rango	28
Mínimo	110
Máximo	138
Suma	1176
Cuenta	10

Grupo de SM

Análisis Colesterol	
Media	201.2
Error típico	10.932782
Mediana	194
Moda	146
Desviación Estandar	42.342481
Varianza de la muestra	1792.8857
Curtosis	-0.7049866
Coeficiente de simetría	0.3200121
Rango	140
Mínimo	146
Máximo	286
Suma	3018
Cuenta	15

Grupo de Control

Análisis colesterol	
Media	154.7
Error típico	7.2050137
Mediana	142
Moda	141
Desviación Estandar	22.784254
Varianza de la muestra	519.12222
Curtosis	-0.605569
Coeficiente de simetría	1.0581247
Rango	62
Mínimo	133
Máximo	195
Suma	1547
Cuenta	10

Figura 7.1: Análisis estadístico en los grupos de estudio.

Como se puede observar en la figura 7.1, se tiene el análisis estadístico para cada uno de los grupos de estudio, en donde se realiza la comparación entre el grupo SM y el grupo de control. A partir de este estudio y los rangos normales, mencionados anteriormente, se puede determinar el grupo de personas las cuales se encuentran en riesgo o ya padecen la enfermedad del SM. Este análisis permitirá el entrenamiento posterior de la ML, donde al transformar el conjunto de datos y realizar una reducción de estos mismos facilitará el correcto funcionamiento de la ML. Cada una de las tablas del estudio estadístico, de los dos grupos de estudio, se encontrarán en el apartado de anexos.

7.1.1. Prueba de Kolmogorov Smirnov

El comprobar la normalidad, en el conjunto de datos de los dos grupos de estudio, se hace fundamental para hallar el *p-valor* por un método de datos normalizados o por un método de datos no normalizados. Por lo cual, se realiza la prueba de Kolmogorov Smirnov. Este proceso se llevó a cabo para cada uno de los items mencionados anteriormente, por medio del siguiente procedimiento:

- Se toma el conjunto de datos, se enumeran cada uno de ellos y se ordenan de mayor a menor, como se observa en la tabla 7.1:

i	**IMC**
1	22.1
2	29.8
3	37.4
4	48.5
5	27.0
6	41.5
7	34.7
8	32.1
9	31.7
10	35.4
11	33.0
12	44.2
13	29.8
14	32.2
15	34.6

Tabla 7.1: Conjunto de datos IMC.

- Se realiza un cálculo percentil para cada uno de los datos, por medio de la siguiente ecuación:

$$p_i = \frac{i}{15}$$

Donde 15 es el número total de datos que posee el grupo de estudio.

Esto da como resultado a la siguiente columna de la tabla llamada p_i, la cual se observa en la tabla 7.2:

i	**IMC**	p_i
1	22.1	0.0666667
2	29.8	0.1333333
3	37.4	0.2
4	48.5	0.2666667
5	27.0	0.3333333
6	41.5	0.4
7	34.7	0.4666667
8	32.1	0.5333333
9	31.7	0.6
10	35.4	0.6666667
11	33.0	0.7333333
12	44.2	0.8
13	29.8	0.8666667
14	32.2	0.9333333
15	34.6	1

Tabla 7.2: Percentil conjunto de datos IMC.

Para comprobar que se realizó dicho procedimiento correctamente, se debe tener en la última fila del conjunto de datos un valor de percentil igual a 1.

- Para hallar el valor tipificado del conjunto de datos, se debe conocer la media y la desviación estándar del conjunto de datos, estos dos valores se hallan por medio de las funciones de Excel **promedio** y **desvest.m** respectivamente. Una vez se tiene estos datos se realiza el cálculo de los valores tipificados Z_i:

$$Z_i = \frac{IMC - media}{desviacion\,estandar}$$

Conociendo cada uno de estos datos, se agregan en la tabla 7.3:

i	**IMC**	p_i	Z_i
1	22.1	0.0666667	-1.823236
2	29.8	0.1333333	-0.67005
3	37.4	0.2	0.4681607
4	48.5	0.2666667	2.1305467
5	27.0	0.3333333	-1.08939
6	41.5	0.4	1.0821952
7	34.7	0.4666667	0.0637965
8	32.1	0.5333333	-0.325591
9	31.7	0.6	-0.385497
10	35.4	0.6666667	0.1686317
11	33.0	0.7333333	-0.190803
12	44.2	0.8	1.4865593
13	29.8	0.8666667	-0.67005
14	32.2	0.9333333	-0.310615
15	34.6	1	0.0488201

Tabla 7.3: Valor tipificado conjunto de datos IMC

- Se halla la probabilidad de Z_i ($P[Z_i]$), como la distribución normal estándar de Z_i, esto se hace por medio de la función de Excel **Distr.Norm.Estand**, una vez se tienen todos los valores de esta probabilidad se consigna en la tabla 7.4:

i	**IMC**	p_i	Z_i	$P(Z_i)$
1	22.1	0.0666667	-1.823236	0.0341338
2	29.8	0.1333333	-0.67005	0.2514131
3	37.4	0.2	0.4681607	0.6801652
4	48.5	0.2666667	2.1305467	0.9834367
5	27.0	0.3333333	-1.08939	0.1379909
6	41.5	0.4	1.0821952	0.8604171
7	34.7	0.4666667	0.0637965	0.5254339
8	32.1	0.5333333	-0.325591	0.3723668
9	31.7	0.6	-0.385497	0.3499346
10	35.4	0.6666667	0.1686317	0.5669568
11	33.0	0.7333333	-0.190803	0.4243399
12	44.2	0.8	1.4865593	0.9314344
13	29.8	0.8666667	-0.67005	0.2514131
14	32.2	0.9333333	-0.310615	0.3780468
15	34.6	1	0.0488201	0.5194687

Tabla 7.4: Probabilidad de Z_i.

- Se realiza el cálculo para la siguiente columna de datos por medio de:

$$Valor\,absoluto = |P(Z_i) - p_i|$$

Dando como resultado el siguiente conjunto de datos, tabulado en la tabla 7.5:

i	IMC	p_i	Z_i	$P(Z_i)$	$\|P(Z_i) - p_i\|$
1	22.1	0.0666667	-1.823236	0.0341338	0.0325329
2	29.8	0.1333333	-0.67005	0.2514131	0.1180798
3	37.4	0.2	0.4681607	0.6801652	0.4801652
4	48.5	0.2666667	2.1305467	0.9834367	0.7167701
5	27.0	0.3333333	-1.08939	0.1379909	0.1953424
6	41.5	0.4	1.0821952	0.8604171	0.4604171
7	34.7	0.4666667	0.0637965	0.5254339	0.0587672
8	32.1	0.5333333	-0.325591	0.3723668	0.1609665
9	31.7	0.6	-0.385497	0.3499346	0.2500654
10	35.4	0.6666667	0.1686317	0.5669568	0.0997098
11	33.0	0.7333333	-0.190803	0.4243399	0.3089934
12	44.2	0.8	1.4865593	0.9314344	0.1314344
13	29.8	0.8666667	-0.67005	0.2514131	0.6152536
14	32.2	0.9333333	-0.310615	0.3780468	0.5552866
15	34.6	1	0.0488201	0.5194687	0.4805313

Tabla 7.5: Valor absoluto de la diferencia entre la probabilidad y el percentil.

- Ahora se halla el mismo valor absoluto, pero para el valor anterior, esto se entiende mejor por medio de la siguiente ecuación:

$$Valor\,absoluto\,dato\,anterior = |P(Z_i) - p_i - 1|$$

Dando como resultado la última columna de datos necesaria para el cálculo del valor **KS**, estos datos se consignan en la tabla 7.6:

i	IMC	p_i	Z_i	$P(Z_i)$	$\|P(Z_i) - p_i\|$	$\|P(Z-i) - p - i - 1\|$
1	22.1	0.0666667	-1.823236	0.0341338	0.0325329	0.03413381
2	29.8	0.1333333	-0.67005	0.2514131	0.1180798	0.18474644
3	37.4	0.2	0.4681607	0.6801652	0.4801652	0.54683183
4	48.5	0.2666667	2.1305467	0.9834367	0.7167701	0.78343675
5	27.0	0.3333333	-1.08939	0.1379909	0.1953424	0.12867573
6	41.5	0.4	1.0821952	0.8604171	0.4604171	0.52708376
7	34.7	0.4666667	0.0637965	0.5254339	0.0587672	0.12543388
8	32.1	0.5333333	-0.325591	0.3723668	0.1609665	0.09429982
9	31.7	0.6	-0.385497	0.3499346	0.2500654	0.18339871
10	35.4	0.6666667	0.1686317	0.5669568	0.0997098	0.03304318
11	33.0	0.7333333	-0.190803	0.4243399	0.3089934	0.24232675
12	44.2	0.8	1.4865593	0.9314344	0.1314344	0.19810105
13	29.8	0.8666667	-0.67005	0.2514131	0.6152536	0.5485869
14	32.2	0.9333333	-0.310615	0.3780468	0.5552866	0.4886199
15	34.6	1	0.0488201	0.5194687	0.4805313	0.41386467

Tabla 7.6: Valor absoluto del dato anterior

- El valor Kolvogorov Smirnov calculado, se halla por medio de las dos últimas columnas de la tabla 7.6, en donde se toma el valor máximo presente en dichas columnas, esto también se puede definir como el error máximo. Por lo cual, para este caso, el error máximo que se tiene es $KS_c = 0{,}783436747$.

- Se debe calcular el valor de Kolmogorov Smirnov por medio de sus tablas, la cual es 7.7:

c_α	α		
Modelo	**0.1**	**0.05**	**0.01**
General	1.224	1.358	1.628
Normal	0.819	0.895	1.035
Exponencial	0.99	1.094	1.308
Weibull n=10	0.76	0.819	0.944
Weibull n=20	0.779	0.843	0.973
Weibull n=50	0.79	0.856	0.988
Weibull $n = \infty$	0.803	0.874	1.007

Tabla 7.7: Valores de Kolmogorov Smirnov

En este caso se toma el valor para él $\alpha = 0{,}05$, debido a que se encuentra en un punto medio. Para hallar el valor de $k(n)$, el cual tiene diferentes ecuaciones para su cálculo, en este caso se utilizará la ecuación para cuando se está comprobando la normalidad del conjunto de datos, esta ecuación es:

$$k(n) = \sqrt{n} - 0{,}01 + \frac{0{,}85}{\sqrt{n}}$$

Donde n es el número de datos que se tienen en el conjunto de datos.

Con este valor de $k(n)$ y con el valor de significancia (c_α), se obtiene el valor de Kolmogorov Smirnov (KS_t), esto por medio de la siguiente ecuación:

$$KS_t = \frac{c_\alpha}{k(n)}$$

Siendo en este caso la división entre 0,895 y 4,082452402:

$$KS_t = \frac{0{,}895}{4{,}082452402} = 0{,}21923097$$

Siendo este valor de KS_t el error máximo permitido, donde si KS_c no supera este valor, se comprueba que el conjunto de datos posee una distribución normal.

Teniendo en cuenta lo anteriormente descrito y que el valor de $KS_c = 0{,}783436747$, se determina que el conjunto de datos no presenta una distribución normal, por lo cual se debe realizar una prueba para valores con distribución no normalizada.

Este mismo procedimiento se realizó para cada uno de los items presentes en el conjunto de datos, tanto para el grupo SM como para el grupo de control. Se muestra el procedimiento con un solo conjunto de datos, en este caso el IMC, por simplicidad. Las tablas de cada uno de los grupos e items, estarán presentes en la sección de anexos.

7.1.2. Prueba U de Mann-Whitney

Como ya se demostró en la sección 7.1.1, el conjunto de datos presenta una distribución normal, por lo que, para hallar el valor p o *p-valor* se debe realizar por medio de un método o prueba para conjuntos de datos no pareados y que no presenten el supuesto de normalidad. Para esto se hace uso de la prueba U Mann-Whitney, en donde se realiza el siguiente procedimiento para cada uno de los items y grupos de estudio:

- Se realiza la comparación entre las dos poblaciones o grupos de estudio, el grupo SM y el grupo de control. El primer grupo posee mayor cantidad de personas en estudio a comparación del segundo, por lo cual, ya se puede observar y hablar de la no paridad de los datos. Esto se establece en la tabla 7.8:

Colesterol (mg/dL)	**Colesterol (mg/dL)**
286.0	175.0
166.0	141.0
247.0	143.0
241.0	140.0
194.0	195.0
146.0	141.0
216.0	151.0
189.0	189.0
237.0	139.0
176.0	133.0
149.0	
146.0	
217.0	
232.0	
176.0	

Tabla 7.8: Valores del colesterol de los grupos de estudio

  En esta tabla 7.8, el grupo que contiene menos cantidad de datos pertenece al grupo SM y los otros datos pertenecen al grupo de control.

- Para el cálculo del estadístico de prueba, que consta de las siguientes ecuaciones:

$$U_1 = n_1 \cdot n_2 + \frac{n_1(n_1+1)}{2} - R_1$$

$$U_2 = n_1 \cdot n_2 + \frac{n_2(n_2+1)}{2} - R_2$$

En donde se deben encontrar los rangos del conjunto de valores de cada uno de los grupos, para esto se realiza una tabla con cada uno de los valores presentes en los dos grupos y se ordenan de menor a mayor, dando como resultado la siguiente tabla 7.9:

Grupo	Colesterol	Rangos
CNT	175.0	12
CNT	141.0	4.5
CNT	143.0	6
CNT	140.0	3
CNT	195.0	18
CNT	141.0	4.5
CNT	151.0	10
CNT	189.0	15.5
CNT	139.0	2
CNT	133.0	1
SM	286.0	25
SM	166.0	11
SM	247.0	24
SM	241.0	23
SM	194.0	17
SM	146.0	7.5
SM	216.0	19
SM	189.0	15.5
SM	237.0	22
SM	176.0	13.5
SM	149.0	9
SM	146.0	7.5
SM	217.0	20
SM	232.0	21
SM	176.0	13.5

Tabla 7.9: Conjunto de datos y su rango

Donde CNT corresponde a los valores del grupo de control y SM corresponde a los

valores del grupo que padecen del SM.

- Una vez se tiene el rango para cada uno de los valores presentes en los dos grupos de estudio, se procede a realizar el cálculo de n_1, que corresponde al tamaño de la muestra del grupo 1, en este caso el grupo SM; por lo cual se tiene que el valor dé $n_1 = 15$. Este mismo paso se realiza para encontrar el valor de n_2, que corresponde al tamaño de la muestra del grupo 2, en este caso el grupo de control; por lo cual se tiene que el valor dé $n_2 = 10$.

- Se debe hallar el valor de R_1 y R_2, los cuales corresponde a la sumatoria de los rangos, tanto del grupo SM como del grupo de control. Se puede interpretar mejor con la siguiente ecuación:

$$R_1 = \sum Rangos\, del\, Grupo\, SM$$

$$R_2 = \sum Rangos\, del\, Grupo\, de\, Control$$

  Dando como resultado que $R_1 = 248{,}5$ y $R_2 = 76{,}5$.

- Conociendo ya los valores de n_1, n_2, R_1 y R_2, se procede a calcular los valores de U_1 y U_2, por medio de las ecuaciones presentadas anteriormente. Dando como resultado lo siguiente:

$$U_1 = 15 \cdot 10 + \frac{15(15+1)}{2} - 248{,}5$$

$$U_2 = 15 \cdot 10 + \frac{10(10+1)}{2} - 76{,}5$$

  Dando como resultado $U_1 = 21{,}5$ y $U_2 = 128{,}5$.

- El valor del estadístico de prueba U, es el valor mínimo presente en U_1 y U_2, por lo cual $U = 21{,}5$.

- Debido a que se tiene un conjunto de datos mayor a 10, se puede calcular una aproximación a la distribución normal estándar por medio de la siguiente ecuación:

$$Z = \frac{U - \frac{n_1 \cdot n_2}{2}}{\sqrt{\frac{n_1 \cdot n_2 (n_1 + n_2 + 1)}{12}}} \sim N(0, 1)$$

- Una vez se tiene esto, se debe definir el valor de significancia, el cual se representa por α y tiene un valor de trabajo, por lo general, de 0.05 ó 5 %.

- Se debe hallar el valor crítico, el cual se define por medio del valor de significacia el cual es del 5 %, lo que indica que se está trabajando con un 95 % de confianza. Este valor es ampliamente conocido, el cual es $Z_1 = 1{,}96$. Para comprobar que este valor es correcto, se puede hallar por medio de la función de Excel **Inv.Norm.Estand**, la cual es la inversa de la distribución normal estándar. La ecuación que define dicho valor es:

$$Z_1 - \frac{\alpha}{2} = Inversa\,de\,la\,distribucion\,normal \cdot [\frac{1 - \alpha}{2}]$$

  Por medio de esta ecuación se comprueba el valor del valor crítico dado previamente.

- La teoría dice que se debe rechazar la hipótesis nula, la cual es $H_0 : P(X > Y) = 0{,}5$, siempre y cuando el valor absoluto de Z sea mayor al valor crítico. Por lo cual, para este caso no se debe aceptar la hipótesis mostrada anteriormente.

- Calculando el *p-valor*, se utiliza la función de Excel **Distr.Norm.Estand.N**, la cual es la distribución normal estándar. Esta función se le aplica a Z, dando como resultado $p - valor = 0{,}001500448$, por medio de este paso también se puede rechazar la hipótesis nula, ya que, el *p-valor* debe ser mayor al valor de significancia.

  Por lo que, se puede afirmar que los niveles de colesterol en $\frac{mg}{dL}$ de las personas del grupo SM son diferentes a los niveles de colesterol de las personas del grupo de control. Estadísticamente, esto se puede expresar por medio de la siguiente hipótesis $H_1 : P(X > Y) \neq 0{,}5$.

  El llegar a estas conclusiones permite saber que el entrenamiento de la ML, no recibirá valores iguales y podrá diferenciar de forma más efectiva entre el grupo

que no padece el SM y el grupo que sí padece el SM. Esto, como se mencionó anteriormente, facilitará el correcto funcionamiento de la ML al momento de arrojar los resultados.

Este mismo procedimiento se realizó para cada uno de los items presentes en el conjunto de datos. Se muestra el procedimiento con un solo conjunto de datos, en este caso los niveles de colesterol, por simplicidad. Las tablas de cada uno de los grupos e items, estarán presentes en la sección de anexos.

7.1.3. Comparación de la mediana en los grupos de estudio

La mediana es el valor que se encuentra en el punto central de una serie o conjunto de datos, por medio de ella se puede identificar que valor corresponde o se encuentra ubicado la posición del conjunto de datos, que presenta un 50 % de lado y lado, es decir, antes y después de este punto en cuestión.

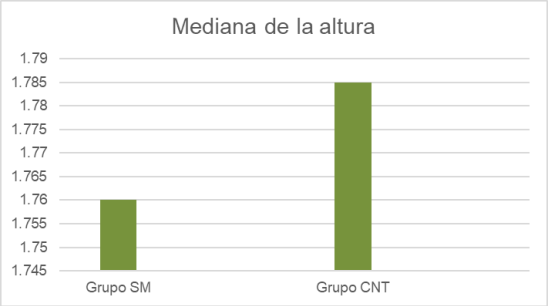
En esta sección 7.1.3, se presentan los valores en mediana de los conjuntos de datos correspondientes a los items o características mencionados en 7.1. Donde, por medio de este, se podrá determinar las diferencias presentes entre los grupos de estudio y cada una de las características y sus rangos, los cuales son los indicadores de la presencia del síndrome metabólico en cada una de las personas en estudio.

Características	Grupo con Síndrome Metabólico	Grupo de Control
Altura	1.76	1.785
Peso	102.3	70.7
IMC	33.006	22.587
Presión Arterial Sistólica	137	116
Presión Arterial Diastólica	88	73
Triglicéridos	194	71
Colesterol HDL	41	46.5
Circunferencia abdominal	108.8	82.3
Ácido úrico	6.28	5.385
ALT	47	29
HOMA	2.58	0.73
Glucosa en tiempo 0	103	96.5
Glucosa en tiempo 30	161	145.5
Glucosa en tiempo 60	165	129
Glucosa en tiempo 90	146	105
Glucosa en tiempo 120	131	96
Insulina en tiempo 0	11	3
Insulina en tiempo 30	70	39
Insulina en tiempo 60	70	28.5
Insulina en tiempo 90	75	29.5
Insulina en tiempo 120	70	28.5

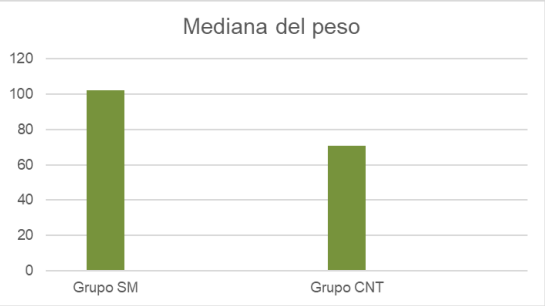
Tabla 7.10: Características de los grupos de estudio, valores dados en mediana.

Con las características presentadas en valores de la mediana, en la tabla 7.10, se puede entrenar la ML de forma más efectiva, ya que, la ML podrá determinar cuáles de estos grupos se encuentran en riesgo, analizando cada una de estas características, buscando aquella que se encuentre fuera de los rangos determinados como normales y así ir conociendo cuáles son las personas en riesgo de padecer o que ya padece el SM.

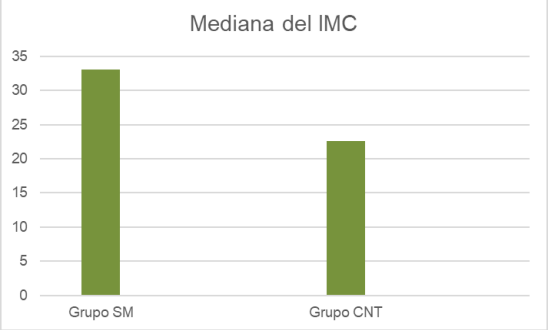
De los valores presentes en la tabla 7.10, se pueden realizar las gráficas correspondientes, las cuales permiten comprobar los *p-valores* que se encontraron en la sección 7.1.2. Estas gráficas se presentan a continuación:



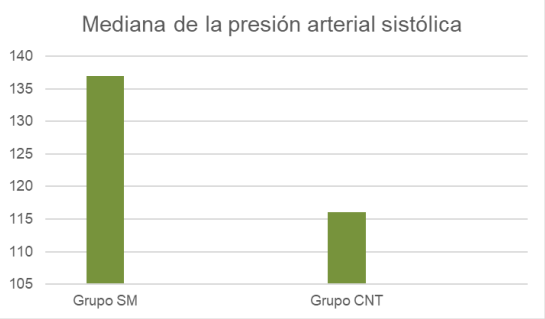
(a) Gráfica de la altura.



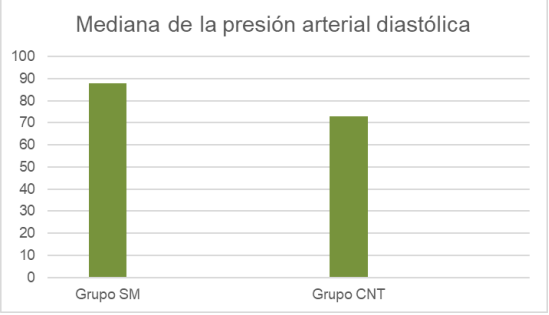
(b) Gráfica del peso.



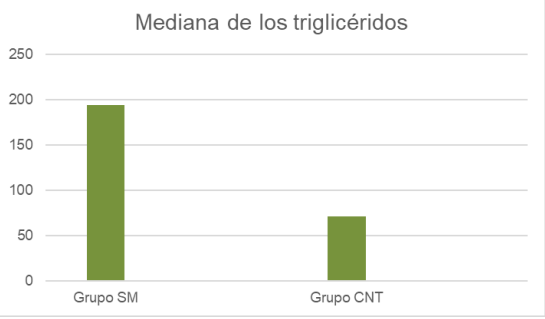
(c) Gráfica del IMC.



(d) Gráfica presión arterial sistólica.

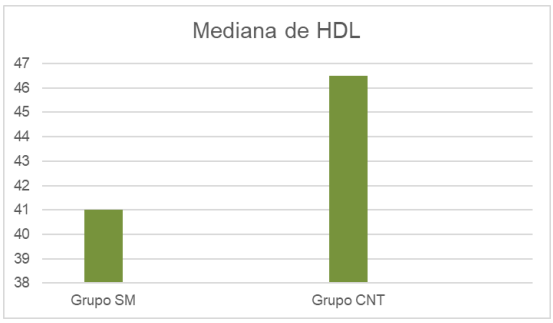


(e) Gráfica presión arterial diastólica.

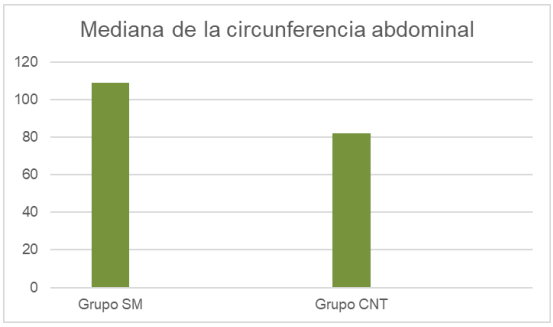


(f) Gráfica de los triglicéridos.

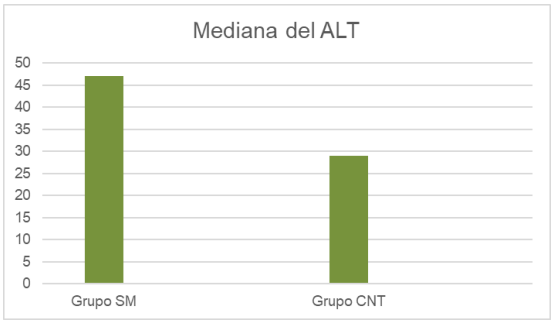
Figura 7.2: Gráficas presentadas en valores de la mediana para seis características.



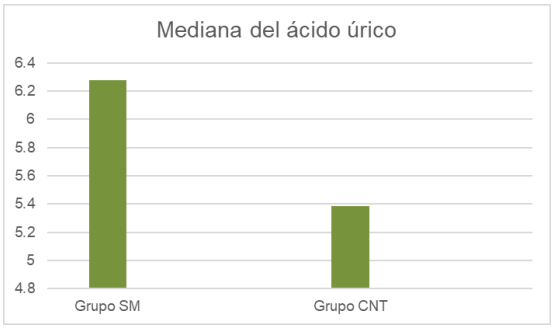
(a) Gráfica colesterol HDL.



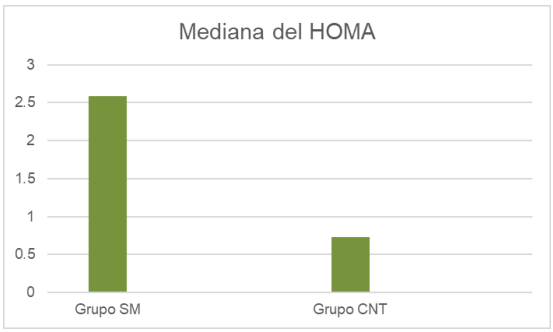
(b) Gráfica circunferencia abdominal.



(c) Gráfica ALT.



(d) Gráfica ácido úrico.



(e) Gráfica HOMA.

Figura 7.3: Gráficas presentadas en los valores de la mediana para cinco características.

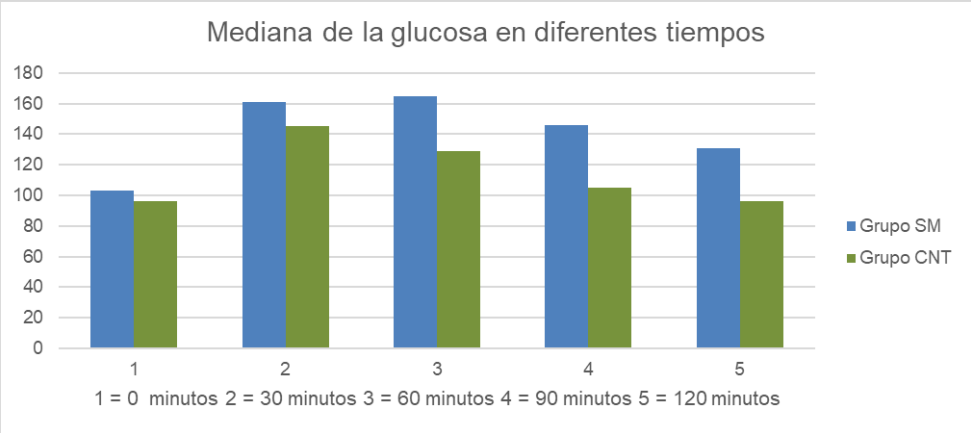


Figura 7.4: Gráfica de la glucosa en diferentes tiempos en valores de la mediana.

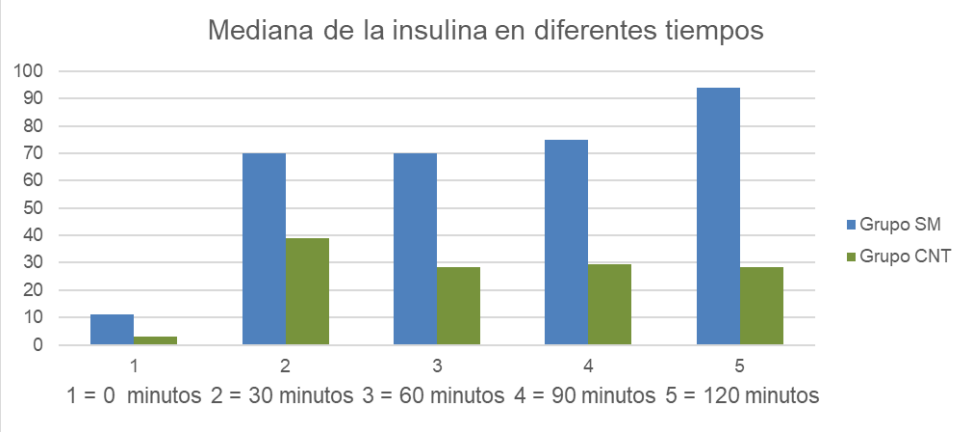


Figura 7.5: Gráfica de la insulina en diferentes tiempos en valores de la mediana.

Como se observa, cada una de las gráficas contiene el valor de la mediana de las características para los dos grupos de estudio (grupo SM y grupo de control). En donde, por lo general, para las características que ponen en riesgo o afectan la salud de las personas, como lo son la circunferencia abdominal, el IMC, las presiones arteriales, entre otras... siempre se presenta un valor más elevado en el grupo padeciente del SM, esto se debe a que estos son los factores o características primordiales, los cuales son indicadores de que se presentan afecciones en la salud de las personas.

7.2. Preprocesamiento de las señales ECG

El preprocesamiento de las señales ECG, se realiza para que al momento de entrenar la ML, esta pueda identificar las componentes esenciales por las cuales se podrá determinar el estado de salud de los grupos de estudio frente al SM. Este preprocesamiento consta de filtrar las diferentes señales que se obtuvieron de los pacientes al hacerles un electrocardiograma, y se encuentran presentes en la base de datos ya mencionada en secciones anteriores; esto con el fin de eliminar factores como el ruido, el cual es una componente no deseada que puede alterar la calidad de la respuesta final. Además, se realiza la identificación de los picos e intervalos de las señales ECG, siendo este un paso fundamental para el entrenamiento de la ML, ya que, se le está facilitando la identificación de estos parámetros al momento de transformar esta información ambigua en datos puntuales, los cuales serán analizados por la ML. Este análisis consta de identificar que rangos posee tanto en el tiempo como en niveles de energía la señal ECG, en donde realizará una comparación entre los rangos normales de cada una de estas componentes y las que se le presentan en la señal de estudio, llegando a determinar las anomalías presentes para dar como resultado si se encuentra alguna afección o signo de alarma que permita determinar si la persona en estudio padece del SM.

Para realizar este preprocesamiento de las señales ECG, se utiliza el lenguaje de programación **Python**, el cual facilita el trabajo por medio de librerías como *numpy* la cual permite crear vectores y matrices que contengan gran cantidad de datos, *matplot* para realizar la graficación de dichas señales, *scipy.signal* para la creación de filtros y *pandas* para el análisis de datos.

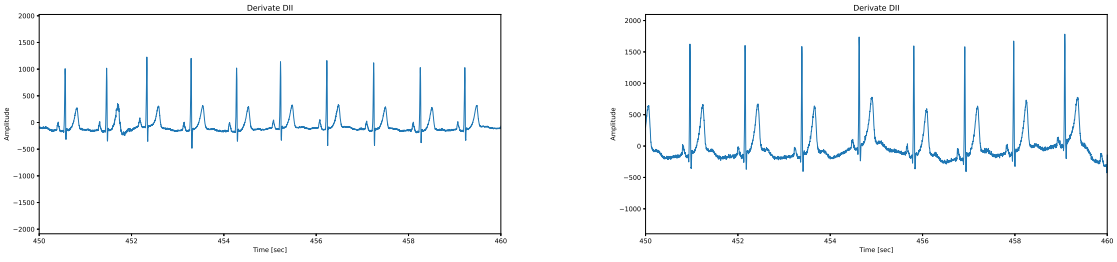
Los pasos a seguir para la transformación de las señales ECG, se pueden dividir en los siguientes:

- Extracción de las señales sin filtrar.
- Señales filtradas por primera vez.
- Señales filtradas por segunda vez.
- Detección de picos y segmentos de las señales.

Se debe tener en cuenta, que para esta sección donde se explicarán cada uno de los pasos y como se llevó a cabo su elaboración, solo se presentarán señales correspondientes a la segunda derivación, esto se debe a que existen estudios donde se demuestra que esta derivación contiene la mayor energía con respecto a las otras, es decir, esta segunda derivación presenta picos más altos respecto a las otras, lo cual implica mejor y mayor visibilidad.

7.2.1. Extracción de las señales ECG sin filtrar

Como primer paso del preprocesamiento de las señales ECG, se extraen dichos elementos de las bases de datos que se encuentran almacenadas en archivos Excel o .csv, se cargan estos datos al código y se establece una frecuencia de muestreo dé $1000\,Hz$. Luego de realizar esta acción, se configura el ambiente de programación para imprimir las derivaciones requeridas, en este caso, como se mencionó anteriormente, será la segunda derivación.



(a) Señal del grupo SM. (b) Señal del grupo de control.

Figura 7.6: Señales ECG sin filtrar correspondientes a la segunda derivación

Como se observa en las figuras 7.6a y 7.6b, las señales ECG presentan una gran cantidad de ruido, lo cual afectará notoriamente la respuesta final, como se mencionó anteriormente. Por lo cual, es preciso eliminar esta componente no deseada por medio de las siguientes etapas de filtrado.

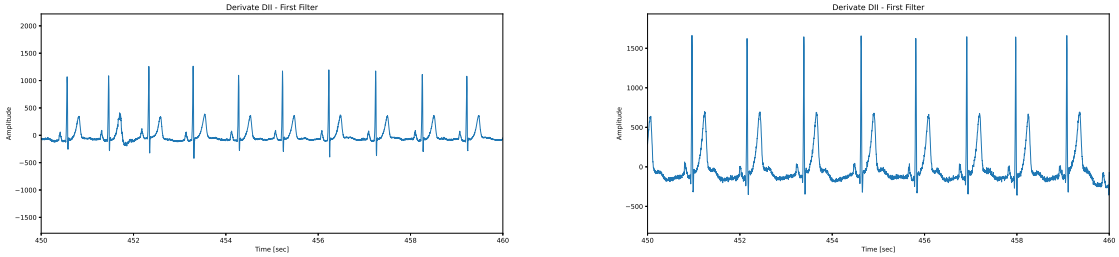
7.2.2. Primer filtrado de las señales ECG

Como ya se observó en el paso anterior, las señales ECG recién extraídas de la base de datos, presentan una gran cantidad de ruido, el cual se debe eliminar, para esto se debe realizar un filtro que elimine las frecuencias por debajo de $0{,}5\,Hz$ y deje pasar toda aquella frecuencia superior a este valor. Teniendo en cuenta, la gran cantidad de filtros que se pueden escoger, se hace uso de un filtro Butterworth de quinto orden, el cual está descrito por la siguiente ecuación:

$$H(s) = \frac{1}{\sqrt{1 + \left(\frac{0{,}5}{1000}\right)^{2 \cdot 5}}}$$

Téngase en cuenta que la frecuencia de muestreo es de $1000\,Hz$.

Una vez se tiene este filtro, se pasa la señal ECG extraída del paso anterior a través de este con el fin de atenuar o eliminar el ruido, como resultado se tiene lo siguiente:



(a) Señal del grupo SM. (b) Señal del grupo de control.

Figura 7.7: Señales ECG filtradas por primera vez correspondientes a la segunda derivación

Se observa en las figuras 7.7a y 7.7b, una atenuación del ruido a comparación de las figuras 7.6a y 7.6b, sin embargo, dicha atenuación no es suficiente y el ruido presente podría generar aún imperfecciones en la respuesta final. Por lo que, se debe realizar una segunda filtración de esta señal, para lo cual se pasa al siguiente paso.

7.2.3. Segundo filtrado de las señales ECG

El paso anterior, aunque atenúo el ruido presente en las señales ECG, no logra proporcionar la suficiente calidad en dichas señales. Por lo cual, se decide aplicar un segundo filtro, el cual eliminará el ruido restante de la señal.

Se opta por la implementación de un filtro FIR pasa bajo, el cual eliminará el ruido de frecuencias superiores a $40\,Hz$, este está representado por la siguiente expresión:

$$h[n] = \sum_{k=0}^{N} \omega[k] \cdot sinc\left(2\pi \frac{40}{1000}\left(n - \frac{22}{2}\right)\right)$$

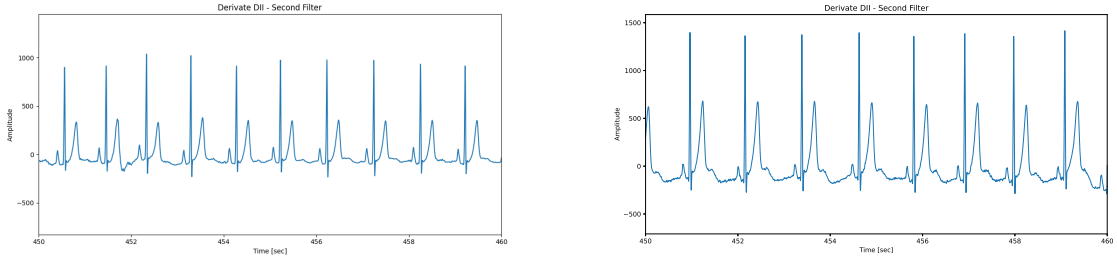
Donde:

- $h[n]$ son los coeficientes del filtro FIR.
- $w[k]$ es la ventana de Hamming aplicada a los coeficientes del filtro, porque este filtro se diseñó con dicha ventana.

- $sinc(x) = \frac{sin(\pi x)}{\pi x}$ es la función utilizada para diseñar los coeficientes FIR.

Téngase en cuenta que la frecuencia de muestreo sigue siendo $1000\,Hz$. Además, el orden del filtro es de 22.

Una vez se tiene este filtro, se pasa la señal ECG extraída del paso anterior a través de este con el fin de atenuar o eliminar el ruido, como resultado se tiene lo siguiente:



(a) Señal del grupo SM. (b) Señal del grupo de control.

Figura 7.8: Señales ECG filtradas por segunda vez correspondientes a la segunda derivación

Como se observa en las figuras 7.8a y 7.8b, se obtiene una señal mucho más limpia a comparación de las señales obtenidas en el paso anterior 7.7a y 7.7b; prácticamente libre de ruido. Facilitando el trabajo con ellas para el paso final y entrenamiento de la ML.

7.2.4. Detección de picos y segmentos de las señales ECG

La detección de picos y segmentos de las señales ECG, se hace esencial para el entrenamiento de la ML, ya que, por medio de la identificación de dichos parámetros, se podrá trabajar de forma más efectiva, eficaz y eficiente, cada una de las señales del corazón. En donde, estos parámetros son los indicadores a estudiar para el hallazgo de anormalidades presentes en cada uno de los pacientes.

Por lo que, se implementa un código el cual contenga los pasos previos de filtrado, para la correcta lectura de los picos y segmentos; y se le agrega una función que determinará cada uno de los segmentos y picos necesarios, los cuales son:

- Pico R.
- Pico P.
- Pico T.

- Segmento QRS.

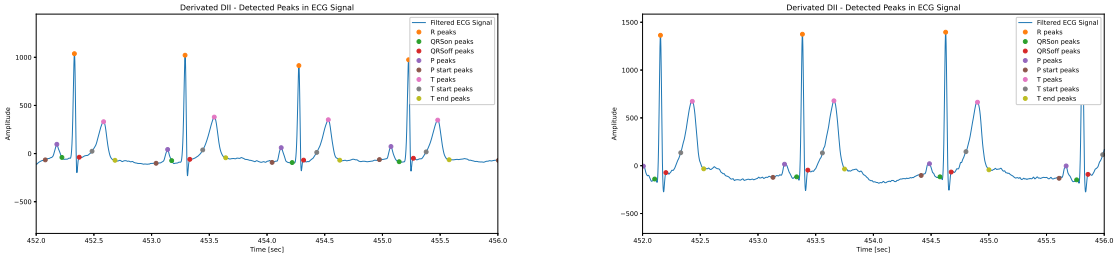
Previamente a la detección de estos parámetros, se realiza un suavizado de la señal por medio de un filtro Savitzky-Golay, para eliminar irregularidades provenientes de las etapas anteriores.

Una vez se realiza este suavizado de la señal ECG, se implementan cada una de las detecciones, las cuales se pueden dividir en pasos o etapas primordiales:

- *Detección de picos R:* Se localizan los picos prominentes con umbrales dinámicos.
- *Detección segmentos QRS:* Se encuentran los mínimos alrededor de cada pico R para delimitar el complejo QRS.
- *Detección de ondas y picos P y T:* Se buscan máximos dentro de intervalos antes y después de QRS_{on} y QRS_{off} respectivamente. Se detectan los inicios y finales de las ondas T y P basados en mínimos locales.

Se debe tener en cuenta que para la detección de estos parámetros y la ubicación exacta de estos puntos, se realiza por medio de un procedimiento matemático, el cual toma un porcentaje del tiempo de muestreo, en mili segundos, y a partir de los arreglos de cada parámetro se determina la ubicación exacta bien sea de inicio o final para los segmentos y la ubicación donde se encuentra el pico respectivo. Estos datos se guardan en un archivo CSV, el cual permite realizar las gráficas de las señales con sus respectivos indicadores.

Por lo tanto, siguiendo estos pasos se tiene como resultado las siguientes señales ECG para los dos grupos de estudio:



(a) Señal del grupo SM. (b) Señal del grupo de control.

Figura 7.9: Detección de picos y segmentos en la segunda derivación de las señales ECG.

7.3. Series de tiempo

Las series de tiempo en las señales ECG son características específicas que se van presentando a lo largo del tiempo, las cuales permiten estudiar, durante un tiempo determinado, la actividad eléctrica del corazón, siendo estas fundamentales para la detección de anomalías o diferentes condiciones cardiovasculares.

Las series de tiempo principales, para el diagnóstico de distintas condiciones cardiovasculares, son:

- La serie RR.
- La serie PR.
- La serie QTc.
- La serie de desviación ST.

Con el fin de realizar una evaluación precisa de la actividad eléctrica del corazón, es esencial obtener información detallada que permita detectar y tratar adecuadamente las patologías. Además del análisis de la VFC a partir de la serie de tiempo del intervalo RR, otros indicadores derivados del ECG, como la duración de las ondas P y T, el segmento ST y el intervalo QT, proporcionan información valiosa [57]-[59]. Un estudio realizado por Magnani et al. reveló que los sujetos con síndrome metabólico presentan un alargamiento de los intervalos PR y otros índices derivados de la onda P, sugiriendo una relación entre el síndrome metabólico y las alteraciones en la electrofisiología auricular [60]. Además, Elffers et al. encontraron asociaciones entre los componentes del síndrome metabólico y otros indicadores derivados del ECG, como el intervalo PR, QT y los ejes de la onda T, en individuos con obesidad [61]. Los sujetos con diabetes o síndrome metabólico mostraron un alargamiento del intervalo QT, representando la repolarización y despolarización de los ventrículos [62].

Para cada una de estas series de tiempo, de los intervalos ya mencionados, se realizó un código en Python por separado, el cual arroja los parámetros que alimentarán a la ML y la representación gráfica de cada una de las series de tiempo, tanto para el grupo de control como para el grupo SM. Se profundizará más a detalle en estos códigos en las siguientes secciones, donde se hablará de las características de las series de tiempo en sus intervalos.

7.3.1. Series de tiempo intervalos RR

Esta serie es la secuencia presentada en los intervalos de tiempo entre los picos R, representando la despolarización ventricular, siendo en este intervalo el encargado de reflejar el ciclo cardíaco completo. Se utiliza para realizar la evaluación de la *variabilidad de la frecuencia cardíaca (VFC)*, el cual es un importante indicador del sistema nervioso autónomo.

La VFC son las fluctuaciones entre los latidos sucesivos en los intervalos de tiempo. En donde, sí se presentan niveles altos de VFC, por lo general, es un indicio de una mejor salud cardiovascular. En cambio, sí por el contrario, se presentan niveles bajos de VFC, esto puede estar asociado a enfermedades cardíacas [63].

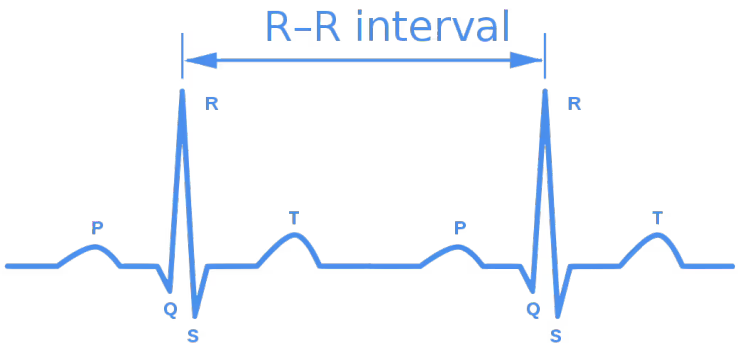


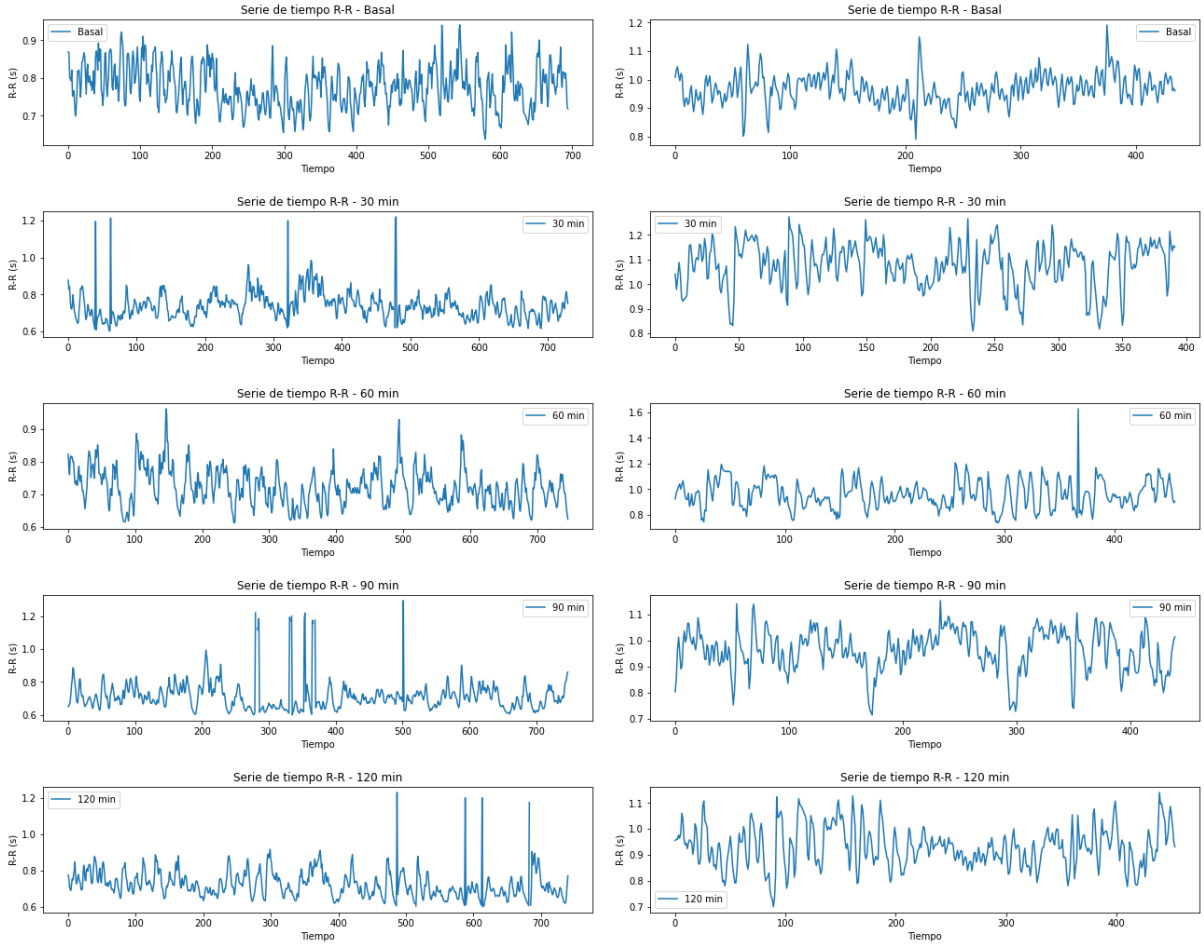
Figura 7.10: Intervalo RR señal ECG [64].

Los valores normales que se manejan en este intervalo son de entre 60 a 100 latidos por minuto, en donde, si se habla de tiempo en segundos, se presentan los siguientes valores para cada una de las pulsaciones, como se muestra en la tabla 7.11:

Frecuencia cardíaca (lpm)	**Intervalo RR (s)**
60	100
70	0.86
80	0.75
90	0.67
100	0.60

Tabla 7.11: Valores normales en latidos por minuto y segundos para el intervalo RR [65].

Para determinar los intervalos presentes en las señales ECG, de las bases de datos que se están estudiando, se realiza un código en Python el cual realiza un proceso de recorte en un rango de tiempo específico en minutos, estos recortes se realizan aproximadamente entre 3 minutos y 12 minutos, luego se almacenan las posiciones de los picos R de las señales ECG, una vez se tienen estas posiciones se calculan los intervalos RR basados en los picos detectados, en donde estos se convierten a segundos utilizando la frecuencia de muestreo y además se calculan las diferencias entre los índices consecutivos de picos. Por último, se obtienen las gráficas de las series de tiempo para el intervalo en cuestión, el cual se muestra en la figura 7.11.



(a) Series tiempo del grupo SM. (b) Series de tiempo del grupo de control.

Figura 7.11: Series de tiempo del intervalo RR en diferentes momentos.

En la figura 7.11, se puede observar las series de tiempo del intervalo RR, presentadas en los diferentes momentos del procedimiento realizado para la detección del SM, tanto para el grupo de control como para el grupo que padece de dicha enfermedad. Las diferencias presentadas entre los dos grupos se debe a las afecciones o anomalías cardíacas que poseen los pacientes del grupo SM, los cuales, como se explicó anteriormente; presentan un deterioro en su salud cardiovascular.

7.3.2. Series de tiempo intervalos PR

Esta serie es la encargada de medir la distancia entre el inicio de la onda P y el complejo QRS en las señales ECG, en donde la onda P es la encargada de representar la despola-

rización de las aurículas y el complejo QRS corresponde a la despolarización ventricular. En el intervalo PR, se observa el tiempo que tarda el impulso eléctrico en viajar desde las aurículas hasta los ventrículos, pasando a través del *nodo auriculoventricular (AV)*.

La serie de tiempo PR analiza la conducción auriculoventricular y detecta los bloqueos AV o las alteraciones en la transmisión del impulso eléctrico. Los valores superiores a 200 ms pueden indicar bloque cardíaco de primer grado, mientras que valores inferiores a 120 ms sugieren la presencia de un camino alterno de excitación entre las aurículas y los ventrículos, o más conocida como el síndrome de Wolff-Parkinson-White [66].

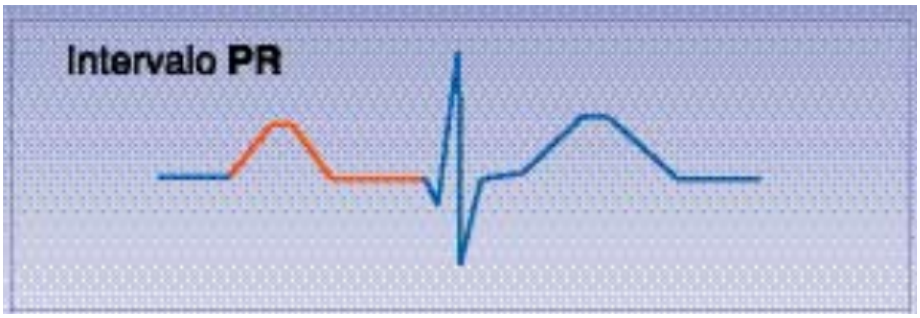
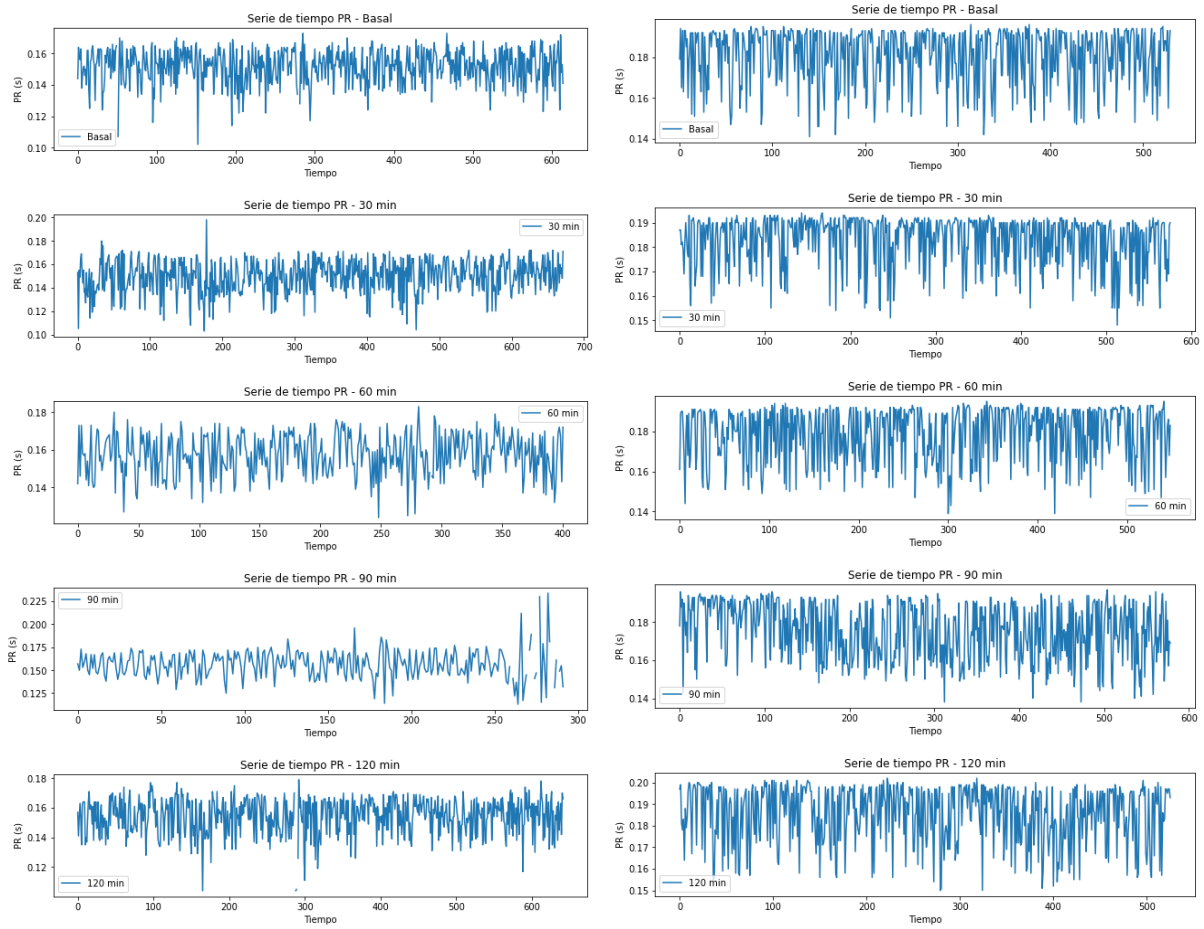


Figura 7.12: Intervalo PR señal ECG [67].

Para determinar este intervalo PR en las señales ECG, se realiza un código en Python, al igual que en el intervalo RR, en donde a partir de las bases de datos de los grupos en estudio se realiza un proceso de corte en intervalos de tiempo de 3 a 12 minutos, convirtiendo estos a índices de muestra teniendo en cuenta la frecuencia de muestreo, luego se realiza la detección del inicio de los picos P y el complejo QRS en el tiempo determinado, con estos dos datos se calcula la duración de la serie del intervalo PR en segundos, este cálculo se hace por medio de la división de la diferencia entre los picos y la frecuencia de muestreo. Por último, se tiene como resultado las gráficas del intervalo PR para los diferentes momentos en las cuales fueron tomadas, para cada uno de los grupos de estudio, como se muestra en la figura 7.13:



(a) Series tiempo del grupo SM. (b) Series de tiempo del grupo de control.

Figura 7.13: Series de tiempo del intervalo PR en diferentes momentos.

En la figura 7.13, se puede observar las series de tiempo del intervalo PR, presentadas en los diferentes momentos del procedimiento realizado para la detección del SM, tanto para el grupo de control como para el grupo que padece de dicha enfermedad. Las diferencias presentadas entre ambos grupos de estudio se deben a las afecciones cardíacas presentes en los pacientes del grupo SM, los cuales tienen series de tiempo por encima de los 200 ms y por debajo de los 120 ms, lo que indica que estos presentan afecciones cardíacas, como se mencionó anteriormente. Por el contrario, los pacientes del grupo de control presentan series de tiempo dentro de los rangos normales.

7.3.3. Series de tiempo intervalos QTc

Esta serie de tiempo va desde el inicio del complejo QRS hasta el final de la onda T, se encarga de medir la duración entre dichos parámetros de las señales ECG. El intervalo QT representa el inicio de la despolarización hasta el final de la repolarización, proceso más conocido como sístole. El valor en función de la frecuencia cardíaca puede presentar inestabilidad al momento de proporcionar un parámetro, haciendo que sea incomparable el intervalo QT entre diferentes ritmos cardíacos, por lo que se hace uso del intervalo QTc, el cual es una versión corregida del parámetro anteriormente mencionado, el cual ajusta el valor en función de la frecuencia cardíaca, haciendo este más estable y permitiendo realizar comparaciones entre diferentes ritmos cardíacos. En este trabajo se utiliza la fórmula de Bazzet ($QT_c = \frac{QT}{\sqrt{RR}}$) para obtener el intervalo QTc.

La serie de tiempo QTc es de gran importancia, ya que permite evaluar la repolarización ventricular [68]. Los valores normales del intervalo en cuestión usando la fórmula de Bazett son [69]:

- Hombres adultos: Entre 350 ms y 450 ms.
- Mujeres adultas: Entre 350 ms y 460 ms.

En donde, si se presenta un valor superior a 450 ms en el intervalo QTc, se puede conducir a arritmias ventriculares y muerte súbita cardíaca.

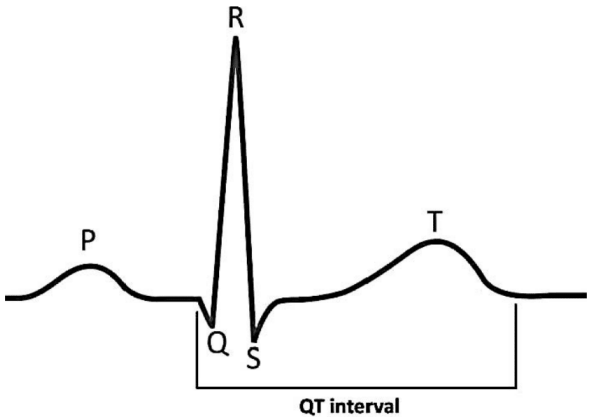


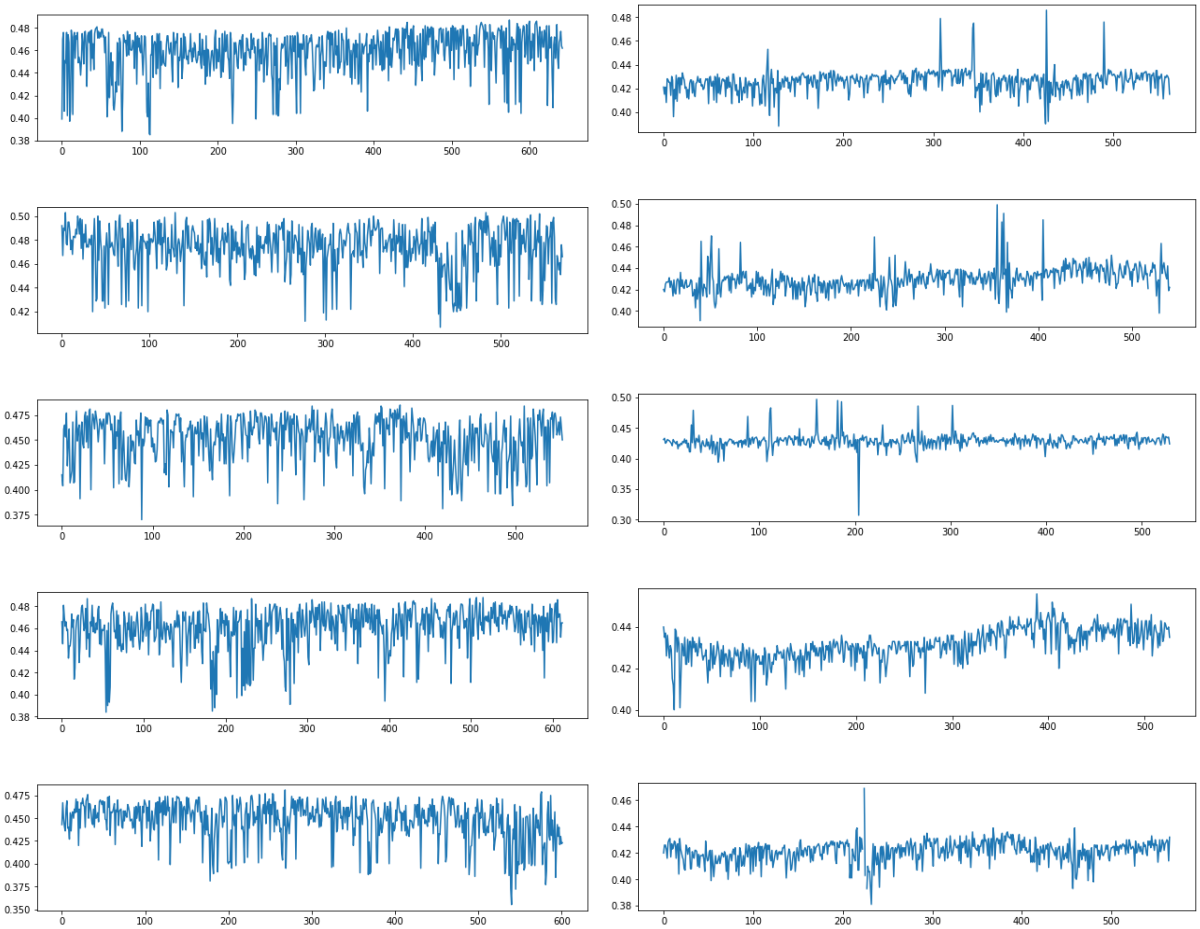
Figura 7.14: Intervalo QTc señal ECG [70].

Para determinar este intervalo PR en las señales ECG, se realiza un código en Python, al igual que en los intervalos anteriores, en donde a partir de las bases de datos de los grupos en estudio se realiza un proceso de corte en intervalos de tiempo de 3 a 12 minutos,

convirtiendo estos a índices de muestra teniendo en cuenta la frecuencia de muestreo. Luego se realiza la detección del inicio del complejo QRS y el final del pico T en el tiempo determinado. Con estos dos datos se calcula la serie de tiempo QTc con la siguiente ecuación:

$$QT_c = \frac{QT}{\sqrt{RR}}$$

En donde $QT = T_{end} - QRS_{on}$. Presentando estos resultados en segundos. Por último, se tiene como resultado las gráficas del intervalo QTc para los diferentes momentos en las cuales fueron tomadas, para cada uno de los grupos de estudio, como se muestra en la figura 7.15:



(a) Series tiempo del grupo SM. (b) Series de tiempo del grupo de control.

Figura 7.15: Series de tiempo del intervalo QTc en diferentes momentos.

En la figura 7.15, se puede observar las series de tiempo del intervalo QTc, presentadas en los diferentes momentos del procedimiento realizado para la detección del SM, tanto para el grupo de control como para el grupo que padece de dicha enfermedad. Las diferencias presentadas entre ambos grupos de estudio, se deben a la presencia del SM en el grupo de pacientes que padecen la enfermedad en estudio, en donde las series de tiempo están por fuera de los rangos normales mencionados anteriormente, sobre todo se puede apreciar que se salen del rango en su límite superior, indicando que pueden existir arritmias ventriculares.

7.3.4. Series de tiempo segmentos ST

Esta serie de tiempo inicia en donde finaliza el complejo QRS, más exactamente en el final de la onda S, hasta el inicio de la onda T. Representa el intervalo entre la despolarización y la repolarización ventricular y suele ser isoeléctrico, ya que todas las células ventriculares se encuentran despolarizadas, este segmento ST puede mostrar elevaciones positivas o elevaciones negativas con respecto al nivel iso-eléctrico, medido en el segmento PR o en el intervalo TP. En este trabajo el nivel iso-eléctrico es obtenido en el segmento PR como el voltaje obtenido 40 ms antes del inicio del complejo QRS. La desviación del segmento ST puede ser causada por isquemias, cambios del ritmo cardíaco y entre otras razones.

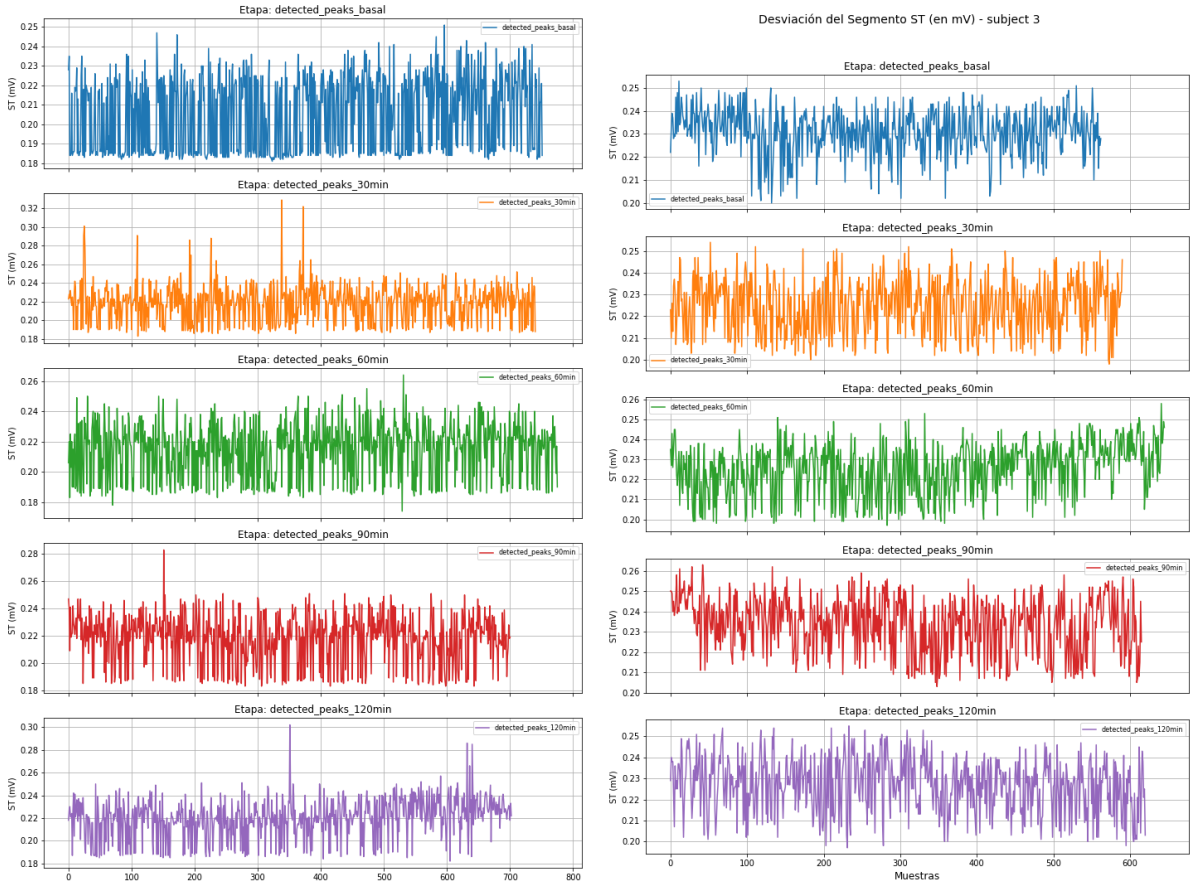
Los valores normales que se presentan en el segmento ST son [71]:

- La duración se encuentra en los 80 ms y los 120 ms.
- *Elevación anormal:* Más de 2 mm (0.2 mV) en hombres y más de 1.5 mm (0.15 mV) en mujeres.
- *Depresión anormal:* Más de 0.5 mm (50 mV).



Figura 7.16: Intervalo ST señal ECG [67].

Para determinar este segmento ST en las señales ECG, se realiza un código en Python, al igual que en los intervalos anteriores, en donde a partir de las bases de datos de los grupos en estudio se realiza un proceso de corte en intervalos de tiempo de 3 a 12 minutos, convirtiendo estos a índices de muestra teniendo en cuenta la frecuencia de muestreo, luego se realiza la detección de la finalización de la onda S o complejo QRS y el inicio de la onda T en el tiempo determinado, con estos dos datos se calcula la duración de la serie del segmento ST en segundos. Por último, se tiene como resultado las gráficas del segmento ST para los diferentes momentos en las cuales fueron tomadas, para cada uno de los grupos de estudio, como se muestra en la figura 7.17:



(a) Series tiempo del grupo SM. (b) Series de tiempo del grupo de control.

Figura 7.17: Series de tiempo del intervalo ST en diferentes momentos.

En la figura 7.17, se puede observar las series de tiempo del segmento ST, presentadas en los diferentes momentos del procedimiento realizado para la detección del SM, tanto

para el grupo de control como para el grupo que padece de dicha enfermedad.

La figura 7.18 muestra la desviación del segmento ST para los grupos de estudio, en diferentes momentos del POTG. Allí, se pueden observar valores similares de desviación del segmento ST. La serie de tiempo ST_{Desv} se forma de la sucesión consecutiva de desviaciones del segmento ST, por lo que se puede analizar la variabilidad de la desviación del segmento ST entre latidos cardíacos.

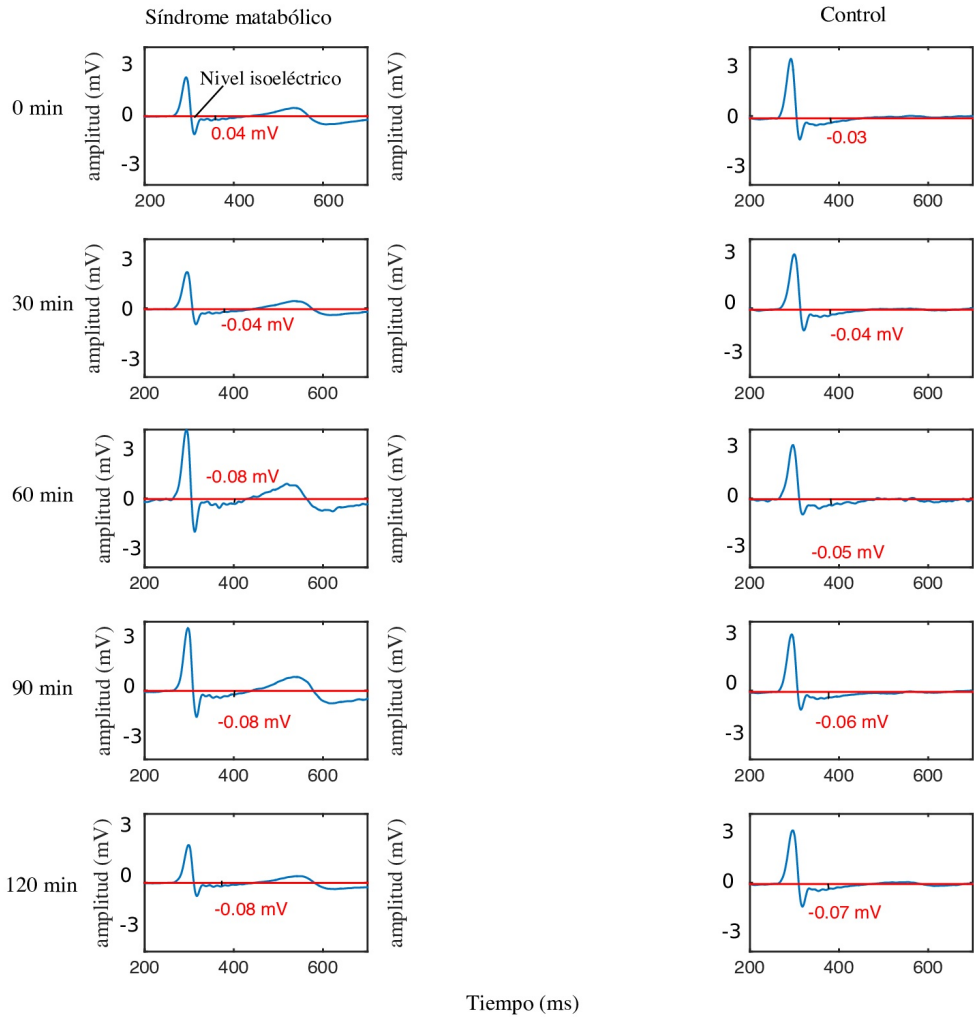


Figura 7.18: Desviación del segmento ST durante el proceso del POTG.

7.4. Parámetros de la señal ECG

La variabilidad de la frecuencia cardíaca (VFC) requiere de una serie de parámetros para ser analizada, estos se dividen en tres categorías:

- Parámetros del dominio del tiempo.
- Parámetros del dominio de la frecuencia.
- Parámetros no lineales.

Estos parámetros analizan diferentes aspectos de la función cardíaca y la regulación autonómica.

7.4.1. Parámetros en el dominio del tiempo

Los parámetros en el dominio del tiempo proporcionan información sobre la variabilidad de los intervalos RR (intervalos entre latidos cardíacos sucesivos) en función del tiempo. Estos parámetros son especialmente útiles para evaluar la función del sistema nervioso autónomo y detectar posibles disfunciones [72], [73].

Como se mencionó anteriormente, este es uno de los grupos de parámetros requeridos para el análisis de la VFC. Se compone de una serie de cálculos, los cuales se presentarán a continuación:

Promedio de los intervalos RR

Este se utiliza para medir el tiempo promedio entre dos latidos consecutivos y es un indicador fundamental de la frecuencia cardíaca [63]. El cálculo se realiza por medio de la siguiente ecuación:

$$Prom_{RR} = \frac{\sum Intervalos\, RR}{Numero\, intervalos\, RR}$$

Desviación estándar de los intervalos RR normales (SDNN)

Es la desviación estándar de todos los intervalos RR, encargada de representar la variabilidad general de estos mismos. Cuando se presenta un mayor SDNN, se habla de una mayor salud cardiovascular [63]. El cálculo se realiza por medio de la siguiente ecuación:

$$SDNN = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(RR_i - \bar{RR}\right)^2}$$

Donde:

RR_i es el intervalo RR i-ésimo.
$\bar{RR}$ es el promedio de todos los intervalos RR.
N es el número total de intervalos RR.

Frecuencia cardíaca promedio (HR)

Calcula la frecuencia cardíaca promedio a partir de los intervalos RR [63]. Esto se hace por medio de la siguiente ecuación:

$$HR = \frac{60}{RR_{\text{promedio}}}\, ppm$$

Desviación estándar del ritmo cardíaco (STD HR)

Mide la desviación estándar del ritmo cardíaco calculada a partir de las series de tiempo RR [63]. Este cálculo se lleva a cabo a partir de la siguiente ecuación:

$$STDHR = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(RR_i - \bar{RR}\right)^2}$$

Donde:

N es el tamaño de la población, en este caso la cantidad de los datos frecuencia cardíaca que se tienen.
x_i es cada valor de la población, es decir, cada valor de la frecuencia cardíaca.
μ es la medida poblacional o el promedio de la frecuencia cardíaca.

Raíz cuadrática media de las diferencias sucesivas de los intervalos RR (rMSSD)

Se mide la variabilidad a corto plazo, por medio de la raíz cuadrada media de las diferencias sucesivas entre los intervalos RR consecutivos [63]. Este cálculo se realiza por medio de la siguiente ecuación:

$$rMSSD = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N-1}(RR_{i+1} - RR_i)^2}$$

Número de intervalos RR sucesivos que difieren en más de 50 ms (NNxx)

Mide el número de intervalos RR consecutivos pares que difieren en más de 50 ms. Para realizar este cálculo o medición se cuentan el número o cantidad de diferencias de pares de intervalos RR consecutivos [63].

Porcentaje de intervalos RR que difieren más de 50 ms (pNNxx)

Mide el porcentaje de intervalos R-R sucesivos que difieren en más de 50 ms respecto al número total de intervalos [63]. Esta medición se hace por medio de la siguiente ecuación:

$$pNN50 = \frac{NN50}{(N-1)} \times 100$$

Índice triangular

Su medición se basa en el análisis de histogramas presentes en los intervalos RR [68]. Este se calcula por medio de la siguiente ecuación:

$$IT = \frac{Intervalos\, RR\, totales}{Altura\, maxima\, histograma}$$

Ancho de la base del histograma de los intervalos RR (TINN)

Se utiliza para medir la dispersión de los intervalos RR por medio del ancho de la base del histograma [68].

Desviación estándar del promedio de los intervalos RR en segmentos de 5 minutos (SDANN)

Su cálculo se realiza por medio de la medición de la variabilidad a largo plazo de la serie de tiempo en segmentos de cinco minutos y además, se calcula la desviación estándar del promedio de cada segmento [66]. Este cálculo se realiza por medio de la siguiente ecuación:

$$SDANN = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(\bar{x}_i - \bar{x}_{\text{total}})^2}$$

Donde:

m es el número total de segmentos de 5 minutos.
$\bar{x}_i$ es la media del segmento i.
$\bar{x}_{total}$ es el promedio global de todos los valores.

Media de la desviación estándar de los intervalos RR en segmentos de 5 minutos (SDNNI)

Se calcula la desviación estándar para cada segmento de 5 minutos y luego se obtiene el promedio de estas [66]. Esto se hace por medio de la siguiente ecuación:

$$SDNNI = \frac{1}{m}\sum_{i=1}^{m}\sqrt{\frac{1}{n_i}\sum_{j=1}^{n_i}(x_{ij} - \bar{x}_i)^2}$$

Donde:

m es el número total de segmentos de 5 minutos.
n_i es el número de observaciones en el segmento i.
x_{ij} es el valor j-ésima observación en el segmento i.
$\bar{x}_i$ es la media del segmento i.

A continuación, se presenta una tabla resumen 7.12 que contiene toda la información descrita anteriormente:

Parámetro	**Unidades**	**Descripción**
RR	ms	Promedio de los intervalos RR.
SDNN	ms	Desviación estándar de los intervalos RR normales.
HR	1/min	Promedio de la frecuencia cardíaca.
STD HR	1/min	Desviación estándar del ritmo cardíaco.
rMSSD	ms	Raíz media cuadrática de las diferencias sucesivas de los intervalos RR.
NNxx	latidos	Número de intervalos RR normales sucesivos que difieren de xx ms (valor por defecto: xx = 50).
pNNxx	%	Porcentaje de intervalos RR sucesivos que difieren de xx ms, es decir NNxx dividido entre el número de intervalos RR totales.
Índice triangular	-	Integral del histograma de los intervalos RR dividido entre la altura del histograma.
TINN	ms	Ancho de la base del histograma de los intervalos RR.
SDANN	ms	Desviación estándar del promedio de los intervalos RR en segmentos de 5 minutos (aplicado en registros de Holter).
SDNNI	ms	Media de la desviación estándar de los intervalos RR en segmentos de 5 minutos (aplicado en registros de Holter).

Tabla 7.12: Parámetros en el Dominio del Tiempo de la VFC [72], [73].

Extracción de los parámetros en cada intervalo del ECG

Los parámetros son extraídos de las series de tiempo de las señales ECG de cada uno de los pacientes. Esto se hace por medio de una implementación de código en Python, el cual procesa y analiza los datos provenientes de los intervalos que componen a la señal ECG. El objetivo principal del código es calcular parámetros estadísticos y almacenar estos resultados para luego poder tener registro de ellos y poder ser utilizados para la creación de los métodos de ML.

El código tiene varias funciones principales para llevar a cabo dicha tarea, en las cuales se realiza el cálculo de cada uno de los parámetros mostrados en esta sección por medio de las ecuaciones dadas. El procesamiento de los datos realiza una iteración sobre los grupos y

los sujetos presentes en cada uno de ellos, en donde se buscan los archivos que contienen la información de cada serie de tiempo de estos mismos, permitiendo así obtener un archivo con los datos de cada uno de los parámetros en los diferentes tiempos en que se realizó la prueba. Vale aclarar que la señal se segmenta en ventanas de tiempo de 5 minutos para analizar tendencias a corto plazo.

7.4.2. Parámetros en el dominio de la frecuencia

El análisis espectral de la variabilidad de la frecuencia cardíaca (VFC) permite identificar y cuantificar los componentes periódicos presentes en el ritmo cardíaco, los cuales se manifiestan como picos en el espectro de potencia.

Como se mencionó anteriormente, este es uno de los grupos de parámetros requeridos para el análisis de la VFC. Se compone de una serie de cálculos, los cuales se presentarán a continuación:

Total Power (TP)

Potencia total en las bandas de frecuencia de 0 a 4 Hz [63]. Para su cálculo se integra la potencia espectral en el rango de 0 a 4 Hz.

Frecuencia pico de la banda VLF

Frecuencia con el pico máximo de la banda VLF entre 0.0033 a 0.04 Hz [68]. Se calcula identificando el pico máximo en el espectro de potencia en esa banda.

Frecuencia pico de las bandas LF y HF

Frecuencias de los picos máximos en las bandas de bajas frecuencias (LF) y altas frecuencias (HF) [68]. Se calcula por medio de la identificación de los picos máximos en las bandas:

- Bajas frecuencias: Entre 0.04 a 0.15 Hz.
- Altas frecuencias: Entre 0.15 a 0.4 Hz.

Potencias en las bandas de frecuencia

Potencias en las bandas VLF, LF y HF mediante integración en sus respectivas bandas [68]. Se calcula por medio de la integración del espectro de potencia en las bandas correspondientes.

Potencia absoluto (log)

Se calcula el logaritmo natural de las potencias absolutas en las bandas VLF, LF y HF [68].

Potencia normalizada LF y HF

Se calcula la potencia de LF y HF en unidades normalizadas [68]. Esto se lleva a cabo por medio de las siguientes ecuaciones:

$$LF_{\text{n.u.}} = \frac{LF}{TP - VLF}$$

$$HF_{\text{n.u.}} = \frac{HF}{TP - VLF}$$

Relación de la potencia LF y HF (LF/HF)

Se calcula la relación de la potencia entre las bandas LF y HF [68]. Esto se lleva a cabo por medio de la siguiente ecuación:

$$LF/HF = \frac{LF}{HF}$$

A continuación, se presenta una tabla resumen 7.13 que contiene toda la información descrita anteriormente:

Parámetro	**Unidades**	**Descripción**
TP	ms^2	Potencia absoluta en las bandas de frecuencia de 0 a 0.4 Hz.
VLF_{peak}	Hz	Frecuencia en la que se observa el pico máximo en la banda de muy bajas frecuencias.
LF_{peak}	Hz	Frecuencia en la que se observa el pico máximo en la banda de bajas frecuencias.
HF_{peak}	Hz	Frecuencia en la que se observa el pico máximo en la banda de altas frecuencias.
VLF	ms^2	Potencia en la banda de muy bajas frecuencias.
LF	ms^2	Potencia en la banda de bajas frecuencias.
HF	ms^2	Potencia en la banda de altas frecuencias.
Potencia absoluta	log	Valores de logaritmo natural de las potencias absolutas en las bandas VLF, LF y HF.
Potencia normalizada	n.u.	Potencia en LF y HF en unidades normalizadas: $\text{LF [n.u.]} = \frac{\text{LF [ms}^2\text{]}}{\text{TP [ms}^2\text{]} - \text{VLF [ms}^2\text{]}}$ $\text{HF [n.u.]} = \frac{\text{HF [ms}^2\text{]}}{\text{TP [ms}^2\text{]} - \text{VLF [ms}^2\text{]}}$
LF/HF	-	Relación de la potencia entre LF y HF.

Tabla 7.13: Parámetros en el Dominio de la Frecuencia de la VFC [74], [75].

Extracción de los parámetros en cada intervalo del ECG

Los parámetros se hallan siguiendo un proceso parecido al utilizado con los parámetros lineales, en donde se tiene un código de Python el cual procesa las series de tiempo de cada uno de los intervalos de la señal ECG de los dos grupos de estudio.

Este tiene las funciones principales que se encargan de calcular cada uno de los parámetros en el dominio de la frecuencia por medio de las ecuaciones presentadas anteriormente. Además, se agregan procesos para detectar y eliminar valores nulos o inválidos como parte del preprocesamiento, permitiendo así no tener parámetros erróneos que afecten posteriormente el desarrollo de la ML.

El cálculo de la densidad espectral de los intervalos RR, se hace a una frecuencia de muestreo de 4 Hz, en donde se definen las bandas de frecuencia de la siguiente manera:

- *VLF:* 0.0033 Hz a 0.04 Hz.

- *LF:* 0.04 Hz a 0.15 Hz.
- *HF:* 0.15 Hz a 0.4 Hz.

Al igual que en los parámetros anteriores se realiza una iteración sobre los grupos y los sujetos presentes en cada uno de ellos, en donde se buscan los archivos que contienen la información de cada serie de tiempo de estos mismos, permitiendo así obtener un archivo con los datos de cada uno de los parámetros en los diferentes tiempos en que se realizó la prueba.

7.4.3. Parámetros no lineales

Estos parámetros permiten analizar las propiedades dinámicas, complejas y caóticas del sistema cardiovascular. Los métodos y teorías utilizadas para llevar a cabo este estudio son la teoría del caos, la fractalidad y la dinámica no lineal. Es de gran importancia llevar a cabo este análisis de dichos parámetros, debido a que permite detectar patrones sutiles y correlaciones que no son evidentes en los enfoques lineales, pudiendo así caracterizar la variabilidad de la frecuencia cardíaca, evaluar enfermedades cardíacas, detectar arritmias y estudiar la regulación autonómica del corazón. En la siguiente tabla 7.14 se muestran aquellos parámetros no lineales que se encuentran a partir de las series de tiempo:

Parámetro	**Unidades**	**Descripción**
SD1	ms	Desviación estándar perpendicular a la línea de identidad en un gráfico de Poincaré.
SD2	ms	Desviación estándar a lo largo de la línea de identidad en un gráfico de Poincaré.
SD1/SD2	-	Relación entre SD1 y SD2.
ApEn	-	Entropía aproximada.
SampEn	-	Entropía muestral.
DFA, α_1	-	Pendiente de las fluctuaciones de corta duración, en el análisis de fluctuaciones sin tendencia.
DFA, α_2	-	Pendiente de las fluctuaciones de larga duración, en el análisis de fluctuaciones sin tendencia.
D2	-	Dimensión de correlación.
RPA	-	Análisis gráfico de recurrencias.
ShanEn	-	Entropía de Shannon.
MSE	-	Entropía multiescala, con factor de escala $\tau = 1, 2, 3, \ldots, 20$.

Tabla 7.14: Parámetros No Lineales de la VFC [72], [73].

Extracción de los parámetros en cada intervalo del ECG

Por último, se tienen los parámetros no lineales, los cuales al igual que en los dos casos anteriores se hallan por medio de un código en Python donde este se enfoca en analizar la variabilidad de la frecuencia cardíaca (VFC) de los dos grupos de estudio y cada uno de los sujetos presentes en estos.

El código consta de varias funciones principales encargadas de calcular los parámetros anteriormente descritos para los parámetros no lineales, esto a partir de la información de las series de tiempo, como se ha venido trabajando anteriormente. Como para estos parámetros no se tienen ecuaciones definidas para el cálculo de cada uno de ellos, se hace uso de diferentes herramientas matemáticas como los son: polinomios, desviación estándar, entropía aproximada, entropía muestral, dimensión de correlación, análisis gráfico de recurrencias, entropía Shannon y entropía multiescala.

Como se ha realizado en los parámetros anteriores se hace uso de una iteración sobre los grupos y los sujetos presentes en cada uno de ellos, en donde se buscan los archivos

que contienen la información de cada serie de tiempo de estos mismos, permitiendo así obtener un archivo con los datos de cada uno de los parámetros en los diferentes tiempos en que se realizó la prueba.

7.5. Machine Learning

Como se mencionó en la sección 5.1, el machine learning se define como "El Machine Learning es una técnica asociada a la detección automática de patrones relevantes dentro de un conjunto de datos." [28, pág. 27]. Se tienen diversas formas por medio de las cuales se instruye al modelo de ML, para el proyecto que se ha venido elaborando se hará uso del aprendizaje supervisado, ya que, el algoritmo aprenderá de cada uno de los datos extraídos de las señales ECG de los grupos y sujetos de estudio. Como parte de este aprendizaje supervisado se tienen múltiples algoritmos, ya nombrados en la sección 5.2.1. Para llevar a cabo la realización del proyecto, se utilizarán dos de las técnicas de clasificación más comunes en el ML, las cuales son:

- Random Forest.
- Support Vector Machine.

Por medio de estos dos métodos se realizará la ML, con el fin obtener los resultados o diagnósticos para cada uno de los sujetos de estudio con los cuales se ha venido trabajando a lo largo del proyecto. Además, se realizará la comparación en cuanto a precisión o exactitud y los recursos consumidos de cada uno de los modelos, para así determinar cuál puede ser la mejor solución para la problemática en cuestión.

7.5.1. Random Forest

Random Forest o Bosques Aleatorios es un modelo de clasificación que recibe este nombre por su particular estructura, en donde este basa su toma de decisiones en las reglas lógicas.

Este es el primer modelo de ML que se realizará para obtener el diagnóstico de los sujetos de estudio, la implementación se realiza por medio de código de Python en donde se hace uso de una librería llamada *Scikit-learn*, su principal característica es que cuenta con los principales algoritmos y funciones unificadas en un solo lugar, permitiendo realizar preprocesamiento, entrenamiento, optimización y validación de forma sencilla [76]; algunos de los algoritmos y/o funciones que maneja son:

- Clasificación.
- Regresión.
- Agrupación.
- Reducción de la dimensionalidad.
- Selección de modelo.
- Preprocesamiento.

Scikit-learn es por excelencia la librería para ML en Python.

Esta librería cuenta con funciones y comandos dedicados a cada uno de los modelos de ML presentes, como se está trabajando con Random Forest se importa de Scikit-learn la clase de *RandomForestClassifier* la cual mejora la precisión y evita el sobre ajuste del modelo, se presenta al momento del aprendizaje en donde el modelo se ajusta a los datos de entrenamiento y es incapaz de predecir correctamente nuevas observaciones [76]. La clase de RandomForestClassifier, presenta esta mejora debido a que proviene directamente del método *ensemble* el cual combina múltiples modelos en un nuevo con el fin de lograr un equilibrio entre bias y varianza, términos que se refieren a cuánto se desvían en promedio las predicciones con respecto a los valores reales y a cuánto cambia el modelo en función de los datos utilizados en su entrenamiento [77].

El entrenamiento se hace por medio de la modificación del *bagging* el cual se basa en el promedio del conjunto de modelos con el fin de disminuir la varianza. Sin embargo, aunque RF es una modificación del método antes mencionado, RF mejora la disminución de la varianza por medio de la selección aleatoria de m predictores antes de evaluar cada división, por lo que se tiene un promedio de $\frac{(p-m)}{p}$ divisiones [77].

La predicción de RF se hace por medio de la media de todas las predicciones de todos los árboles que lo conforman, ejemplificando este proceso se puede entender mejor. Se tiene el siguiente valor de datos:

id	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Y	10	18	24	8	2	9	16	10	20	14
X	...	...	...	...	...	...	...	...	...	...

Tabla 7.15: Ejemplo de predicción [77]

Donde Y corresponde a las variables de respuesta y X a los predictores.

Si se siguen las rutas de la siguiente figura 7.19:

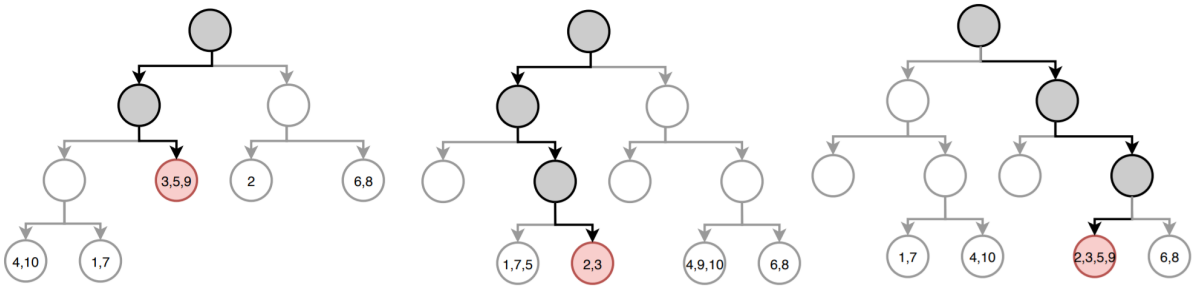


Figura 7.19: Predicción RF ejemplo [77].

La figura 7.19 muestra la ruta de decisiones tomada por un modelo RF, en su destino final muestra la información de las posiciones de id tomadas, es decir, el primer árbol tomó una ruta que pasó por los valores 24, 2 y 20, estos valores se toman en cuenta para realizar la predicción para este árbol, el mismo proceso se sigue con los otros dos árboles de la figura 7.19, teniendo como resultado las siguientes predicciones:

$$y_{arbol_1} = \frac{24 + 2 + 20}{3} = 15{,}333$$
$$y_{arbol_2} = \frac{18 + 24}{2} = 21$$
$$y_{arbol_1} = \frac{18 + 24 + 2 + 20}{4} = 16$$

Estos son los resultados de cada una de las predicciones individuales por árbol, conociendo estos valores se puede hallar la predicción final del modelo, siendo esta lo siguiente:

$$\mu = \frac{15{,}333 + 21 + 16}{3} = 17{,}4$$

De esta manera es como se halla comúnmente tanto las predicciones individuales por árbol creado y la predicción final de todo el modelo en su conjunto.

Como se ha mencionado, existen los predictores los cuales son los encargados de mejorar la capacidad predictiva, sin embargo, como se tienen múltiples árboles se cuentan con estrategias que permiten definir cuál de los múltiples predictores tiene mayor importancia, estas son:

- Importancia por permutación.
- Impureza de nodos.

En el proyecto se hará uso de la segunda estrategia lista, en donde se cuantifica el incremento total en la pureza de los nodos debido a divisiones en las que participa el predictor. Este se calculó por medio de la cada una de las divisiones de los árboles, en donde se registra el descenso obtenido en la medida utilizada como criterio de división, en este caso se realiza por medio del índice de Gini. Dando como conclusión que entre mayor sea este valor medio, se tiene mayor contribución del predictor en el modelo de RF [77].

Teniendo en cuenta todos estos conceptos se puede iniciar con el ajuste del modelo, como se hace uso de la librería Scikit-learn se presentan una lista de hiperparámetros que se encuentran en la clase *RandomForestClassifier*, lo cuales son:

- N_estimator.
- Criterion.
- Max_depth.
- Min_samples_split.
- Min_samples_leaf.
- Min_weight_fraction_leaf.
- Max_features.
- Max_leaf_nodes.
- Min_impurity_decrease.
- Bootstrap.
- Oob_score.
- N_jobs.
- Random_state.
- Verbose.
- Warm_start.

- Ccp_alpha.
- Max_samples.

En el caso del proyecto solo se hará configuración de los hiperparámetros n_estimator, max_features y random_state, los cuales se definen como [77]:

- **N_estimator:** Número de árboles del modelo.
- **Max_features:** Número de predictores de cada división.
- **Random_state:** Conjunto de valores que toma el modelo para realizar la o las predicciones.

Con estos valores se puede crear el modelo RF, sin embargo, se requiere de una optimización de ellos para definir cuáles son los valores que arrojarán mayor exactitud al momento de realizar la toma de decisiones y obtener el diagnóstico final. Para esto se realiza el código correspondiente en Python que analiza el comportamiento de estos hiperparámetros con respecto al aumento, bien sea del número de árboles o del número de predictores, obteniendo como resultado lo siguiente:

Optimización de los hiperparámetros

Para hallar cuál es el número de árboles más óptimo del modelo, según los datos de los parámetros hallados en la sección 7.4, se realizan dos métodos:

- **Validación cruzada:** La cual es una técnica que evalúa el modelo de forma automática, dividiendo los datos en subconjuntos y entrenándolo en algunas de dichas partes mientras se evalúa el resto.
- **Out-of-Bag error:** Este tiene el mismo objetivo de la validación cruzada con la diferencia de que no se requiere de separación de los datos y estima el error del modelo.

Sabiendo esto, se realiza el hallazgo tanto del número de árboles como del max_features, teniendo como resultados las siguientes gráficas y valores:

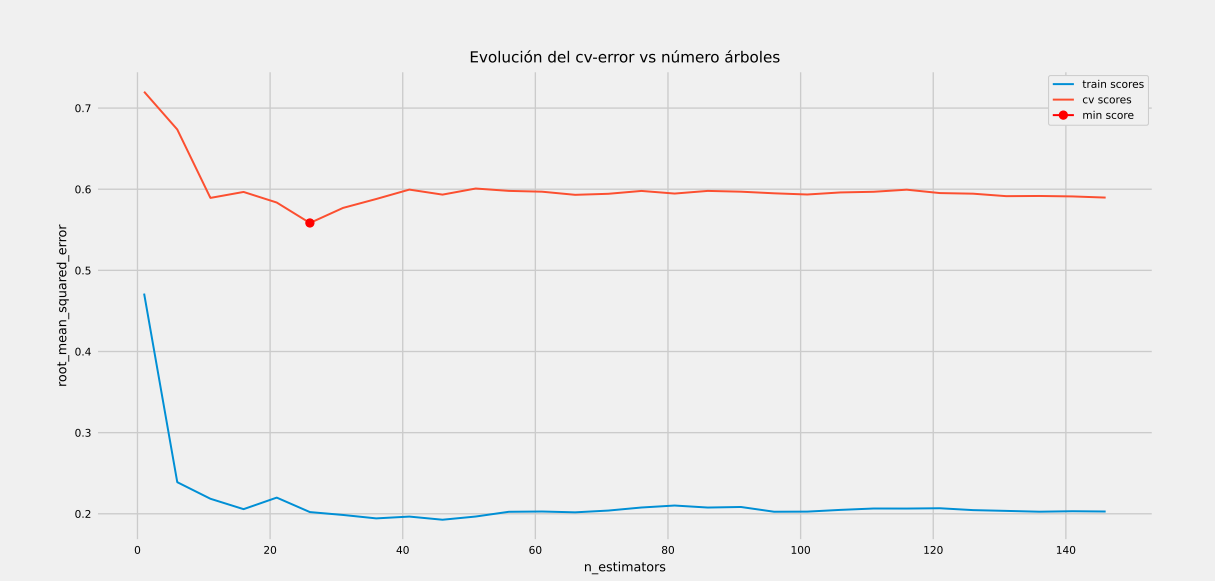


Figura 7.20: Validación cruzada vs Número de árboles.

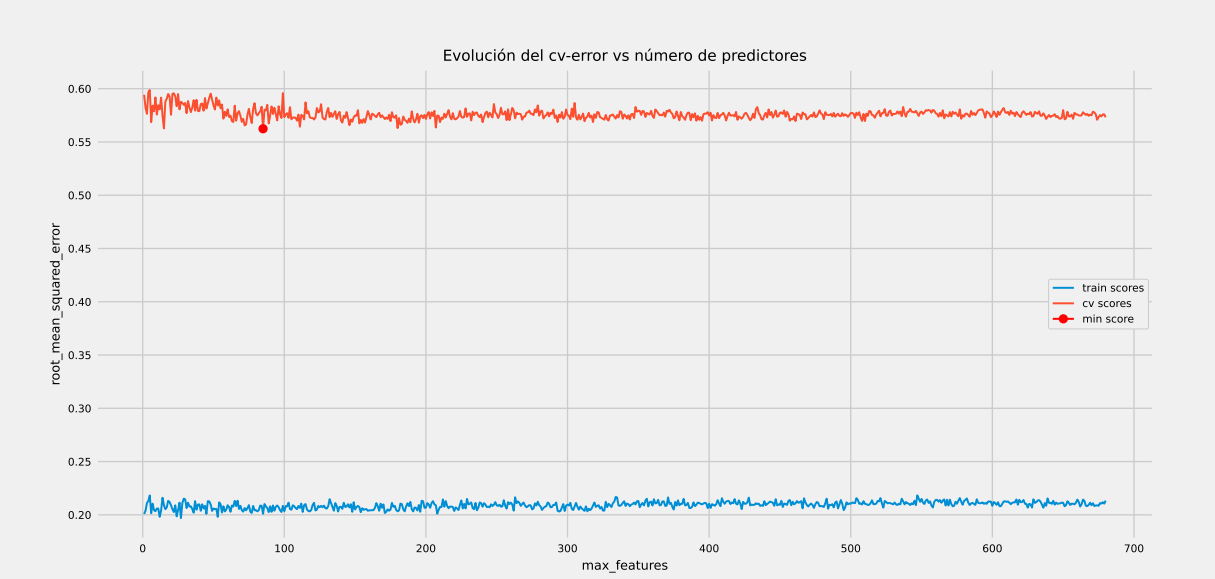


Figura 7.21: Validación cruzada vs Número de predictores.

Por medio del método de validación cruzada, se obtiene que el valor óptimo para n_estimators es de 26 y para max_features es de 85.

Ahora bien, se tiene el método de Out-of-Bag error, el cual arroja los siguientes resultados:

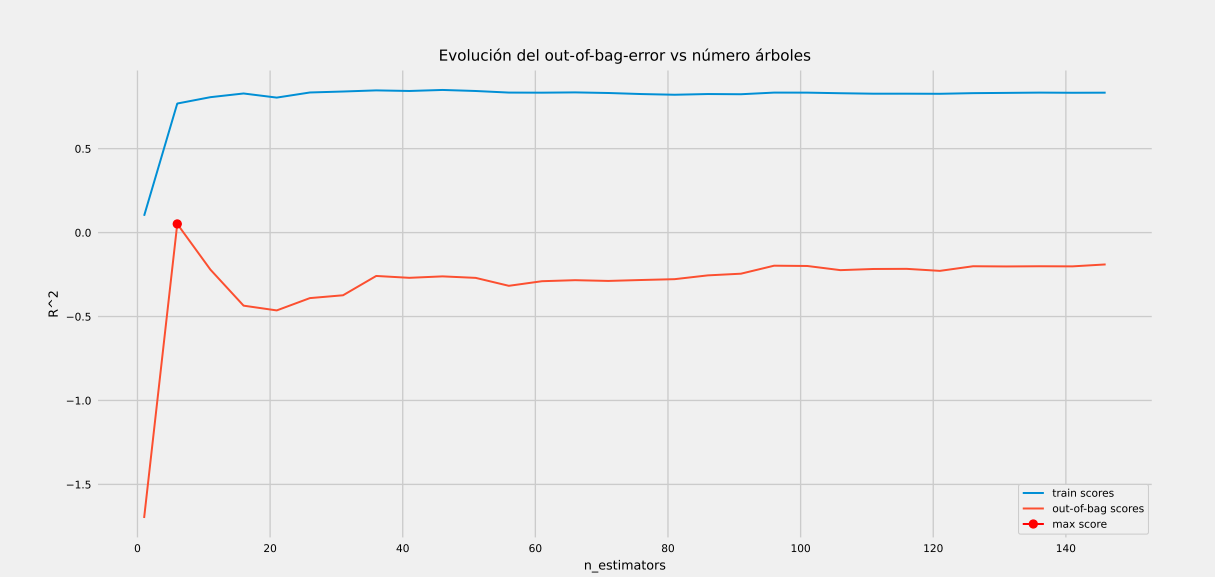


Figura 7.22: Out-of-Bag error vs Número de árboles.

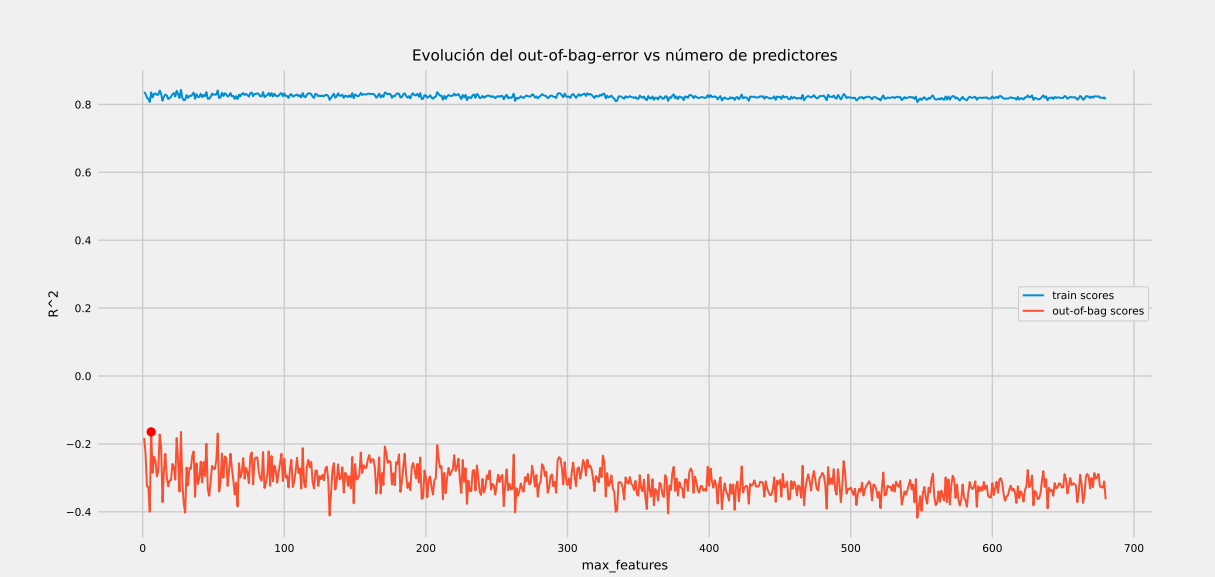


Figura 7.23: Out-of-Bag error vs Número de predictores.

En donde se obtiene que el valor óptimo para n_estimators es de 6 y para max_features es de 6.

Estos hiperparámetros se pueden utilizar en la clase *RandomForestClassifier*, sin embargo, no se debe realizar la búsqueda secuencial de estos, debido a que cada uno de los hiperparámetros interactúan con los demás. Para ello se usó del método *Grid Search*, el cual se utiliza para ajustar los hiperparámetros de forma automática con la finalidad de encontrar la combinación óptima que maximice el rendimiento del modelo, evaluando todas las combinaciones posibles de un conjunto predefinido de hiperparámetros.

Optimizando los hiperparámetros mencionados anteriormente, por medio del método de Grid Search combinando con la validación cruzada, se tiene que los valores óptimos para realizar el RF con los datos de los parámetros hallados en la sección 7.4 son:

- N_estimators: 150.
- Max_features: 5.
- Max_depth: 3.

Estos son los valores de los hiperparámetros que se utilizarán para la creación del modelo de RF.

Hace falta el hallar el valor de un hiperparámetro (random_state), para este se crea un código de Python aparte en donde se recorre desde 0 hasta 100 el valor del random_state (téngase en cuenta que este valor debe ser un número entero) haciendo uso de los tres hiperparámetros antes encontrados. Los resultados de algunos de los valores recorridos se pueden observar en la siguiente figura 7.24:

```
Probando random_state = 29...
Datos cargados correctamente.
Random state: 29
Tamaño conjunto entrenamiento: 18
Tamaño conjunto test: 6
Exactitud del modelo: 0.83

Probando random_state = 30...
Datos cargados correctamente.
Random state: 30
Tamaño conjunto entrenamiento: 18
Tamaño conjunto test: 6
Exactitud del modelo: 0.50

Probando random_state = 31...
Datos cargados correctamente.
Random state: 31
Tamaño conjunto entrenamiento: 18
Tamaño conjunto test: 6
Exactitud del modelo: 0.17

Probando random_state = 32...
Datos cargados correctamente.
Random state: 32
Tamaño conjunto entrenamiento: 18
Tamaño conjunto test: 6
Exactitud del modelo: 0.50
```

Figura 7.24: Valores de random state y exactitud del modelo.

Como se puede observar en 7.24, el hiperparámetro varía desde 29 hasta 32, en este caso, en donde el valor de 29 tiene una exactitud del 83 %, este es el valor más óptimo para dicha variable, por lo cual es el escogido para complementar a los 3 hiperparámetros hallados anteriormente.

Con los valores óptimos de los hiperparámetros hallados, se procede a realizar la construcción del modelo RF.

Construcción del modelo RF

La construcción del modelo RF se lleva a cabo por medio un código en Python, el cual hace uso de las siguientes librerías:

- **Pandas:** Para la lectura de los archivos csv y el manejo de DataFrames.
- **Numpy:** Para el manejo de datos en forma de arreglos o matrices.

- **Scikit-learn:** Para el uso de algoritmos y funciones que permiten la creación de la ML.

Este desarrollo se puede dividir en 10 secciones específicas, las cuales son:

1. **Lectura e importación de datos:** Se carga el archivo csv con todos los parámetros extraídos de las series de tiempo, por medio de la librería de pandas se realiza la lectura de estos y se convierten en un DataFrame el cual será utilizado, entrenará al modelo y le permitirá hacer tanto pruebas como predicciones.

2. **Verificación de existencia del archivo:** Se realiza la comprobación de que el archivo seleccionado se encuentre en las carpetas seleccionadas.

3. **Separación de los parámetros:** Los parámetros ya almacenados en el DataFrame, en una variable llamada *data*, se separan en dos variables **X** la cual almacenará todos los datos de los parámetros e **Y** la cual almacenará la columna final que corresponde al diagnóstico de los pacientes.

4. **Comprobación de los valores:** Se comprueba de que los valores presentes en el DataFrame no contengan ningún valor nulo que pueda afectar al entrenamiento y predicción del modelo.

5. **División de los datos en conjuntos de entrenamiento y prueba:** Se dividen los datos almacenados en **X** e **Y** en conjuntos tanto de prueba como de entrenamiento, es decir, se tendrá un número de datos de **X** que se utilizarán para el entrenamiento del modelo y otros para realizar las pruebas de testeo, así mismo con los datos almacenados en la variable **Y**. Esto se hace por medio de la función *train_test_split* en donde se asigna un 25 % del conjunto de datos para pruebas y el 75 % restante para entrenamiento, estos valores se eligen de esta manera, ya que, se tiene un total de 24 sujetos de estudio, con lo cual el 25 % de ellos es exactamente un número entero que en este caso es 6, si se toma otro valor diferente se puede llegar a tener valores no exactos que afectarán a la predicción del modelo y a su vez si se toma un porcentaje más elevado para las pruebas se tendría muy pocos datos de entrenamiento afectando directamente la exactitud del modelo.

6. **Creación del modelo:** Por medio de la clase o función *RandomForestClassifier* se elabora el modelo, en donde se agregan los cuatro hiperparámetros hallados previamente.

7. **Entrenamiento del modelo:** El modelo se entrena por medio de los datos separados en de las variables **X** e **Y** que en este caso se almacenaron en nuevas variables llamadas **X_train** e **Y_train**, una vez el modelo analizó y aprendió de estos parámetros está listo para realizar predicciones.

8. **Predicciones del modelo:** Se asigna a la variable **y_pred** los datos de la variable **X_test** que son los datos de prueba seleccionados previamente. Acá el modelo realiza la predicción con base en lo aprendido en la sección anterior y almacena sus resultados en la variable correspondiente.

9. **Evaluación del modelo:** Se evalúa el modelo con el fin de hallar la exactitud de este mismo, esto se hace por medio las variables **y_test** e **y_pred**, a las cuales se les realiza un promedio de los resultados obtenidos en cada una de sus interacciones.

10. **Excepciones y errores:** Se crean alertas para el caso donde se presenten problemas con el archivo csv o donde se tengan datos nulos.

Una vez se explican las secciones que componen a la creación del modelo y se crea este mismo, se procede a observar los resultados obtenidos.

7.5.2. Support Vector Machine

El segundo modelo de ML que se realizará para predecir la enfermedad del SM será SVM. Al igual que en el modelo anterior, este será desarrollado por medio de un código elaborado en Python en donde se hace uso de la librería, ya mencionada, *Scikit-learn*.

Este modelo cuenta con una particularidad, ya que es un algoritmo de clasificación múltiple y regresión desarrollado en un hiperplano de N dimensiones. Un hiperplano se define como subespacio del plano y afín de dimensiones $p - 1$ [78].

La clasificación binaria haciendo uso del hiperplano se puede hacer de dos maneras:

- **Casos perfectamente separables linealmente:** Son aquellas ocasiones en donde las observaciones o datos están perfectamente separadas linealmente entre sí.
- **Casos cuasi-separables linealmente:** Son aquellas ocasiones en donde las observaciones o datos no presentan un comportamiento el cual permita hacer la separación lineal perfecta de los grupos de datos, lo cual es más común de presentarse en la realidad, por lo que no existe forma de separar dichas observaciones por medio de un hiperplano.

Los datos de los parámetros extraídos en la sección 7.4 no presenta observaciones que se puedan separar linealmente debido a su gran cantidad de datos y a la distribución de ellas en el plano, por lo que se hace uso de la segunda clasificación binaria. El hiperplano creado para llevar a cabo este tipo de clasificación se denomina como Support Vector Classifier (SVC), el cual permite que el modelo sea más robusto, tenga mayor capacidad predictiva

al momento de agregar nuevas observaciones y presente menos problemas de *overffiting*. Esto se hace por medio del uso de un margen más amplio de clasificación, permitiendo que algunas observaciones estén ubicadas del lado incorrecto del margen o del hiperplano. Para esto es fundamental hacer uso del hiperparámetro **C**, el cual es el encargado de controlar el número y la exigencia de las violaciones del margen, entre más alto es el valor de **C** permitirá que menos observaciones incumplan con la norma establecida por el margen, en cambio, mientras más pequeño sea el valor de **C** menos penalizaciones a los errores se tendrán, por lo que **C** es el encargado de controlar el balance entre el bias y la varianza [78].

El SVC pierde precisión cuando se presentan casos en donde las observaciones no pueden ser separadas linealmente, esto se puede evitar expandiendo las dimensiones del espacio inicial. Las observaciones puede que no sean separables linealmente en el espacio inicial, sin embargo, esto no quiere decir que no sea puedan separar en otros espacios más grandes, esto se puede ver ejemplificado en la siguiente figura 7.25:

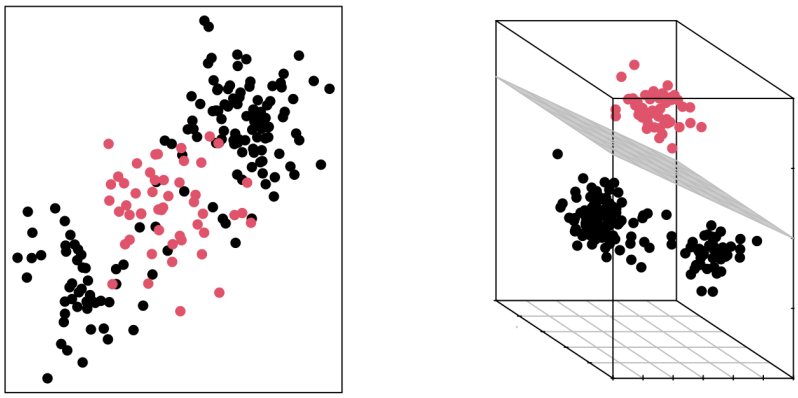


Figura 7.25: Separación de las observaciones haciendo aumento de los espacios [78]

La dimensión de un conjunto de datos u observaciones se puede transformar combinando o aumentando cualquiera de sus dimensiones. Para determinar cuál de estas múltiples posibles dimensiones es la adecuada se debe tener en cuenta el término de *kernel*, el cual "es una función que devuelve el resultado del producto escalar entre dos vectores realizado en un nuevo espacio dimensional distinto al espacio original en el que se encuentran los vectores" [78]. Existen diferentes tipos de kernels, los más utilizados son:

- Kernel lineal.
- Kernel polinómico.

- Gaussian Kernel.

Para determinar cuál es el kernel correcto a utilizar en el modelo de SVM, se debe realizar la prueba por medio de la validación cruzada (la cual también se utilizó en el modelo RF). Allí, se determinará el tipo de kernel que separará de forma correcta las observaciones o datos para la correcta clasificación de estos mismos.

En Python la clase *SVC*, que es la que se utiliza para la creación del modelo SVM, tiene varios hiperparámetros dentro de sí al igual que el modelo RF estos son:

- C.
- Kernel.
- Degree.
- Gamma.
- Class_weight.
- Probability.
- Tol.

En el caso del proyecto solo se hará uso y se modificarán los hiperparámetros de:

- C.
- Kernel.
- Degree: Grado del polinomio para cuando se usa un kernel polinómico.
- Random_state.

Con estos hiperparámetros se puede crear el modelo SVM; sin embargo, como en el modelo de RF, los valores que se le asignarán a estos elementos se deben hallar por medio de un código de Python en donde halla los hiperparámetros óptimos para obtener un mejor diagnóstico.

Optimización de los hiperparámetros

La optimización de los hiperparámetros se hace por medio del método *Grid Search CV*, es decir, el método ya utilizado en la optimización del modelo RF y además se le combina el método de validación cruzada. Dando como resultado los siguientes hiperparámetros para el modelo SVM:

- C: 100.
- Kernel: Polinómico.
- Degree: 2.

Esto indica que se hará uso de un kernel polinómico de grado 2, el cual está descrito por la siguiente ecuación general:

$$K(x, x') = (x \cdot x' + C)^d$$

Donde si $d > 1$, se generan límites de decisión no lineales, se tiene como recomendación el no usar un valor de d mayor a 5 para evitar problemas de *overffiting*.

Una vez se conocen estos valores óptimos para los hiperparámetros, se debe hallar el valor óptimo para el random_state. Esto se hace por medio de un código en Python que es similar al utilizado en la sección 7.5.1, en donde se recorren los valores de random_state desde 0 hasta 100 con el fin de hallar aquel número entero que presente la mayor exactitud en el modelo al momento de ser evaluado, teniendo en cuenta los 3 hiperparámetros hallados anteriormente. Los resultados de algunos de los valores recorridos se pueden observar en la siguiente figura 7.26:

```
Probando con random_state=4
Exactitud: 0.67

Probando con random_state=5
Exactitud: 1.00

Probando con random_state=6
Exactitud: 0.67

Probando con random_state=7
Exactitud: 0.83

Probando con random_state=8
Exactitud: 0.50

Probando con random_state=9
Exactitud: 0.83

Probando con random_state=10
Exactitud: 0.67

Probando con random_state=11
Exactitud: 0.67
```

Figura 7.26: Valores de random state y exactitud del modelo.

Como se observa en la figura 7.26, el hiperparámetro varía desde 4 hasta 11 en este caso, en donde el valor de 5 tiene el 100 % de precisión. Este valor es el más óptimo para dicha variable, por lo cual es el elegido para completar y acompañar a los 3 hiperparámetros hallados previamente.

Con los valores óptimos de los hiperparámetros hallados, se procede a realizar la construcción del modelo SVM.

Construcción del modelo SVM

La construcción del modelo RF se lleva a cabo por medio un código en Python, el cual hace uso de las siguientes librerías:

- **Pandas:** Para la lectura de los archivos csv y el manejo de DataFrames.
- **Scikit-learn:** Para el uso de algoritmos y funciones que permiten la creación de la ML.

El desarrollo de este modelo tiene métodos o funciones similares al modelo RF, por lo cual se repiten la gran mayoría de ellas. En este caso se divide en 10 secciones específicas, las cuales son:

- Lectura e importación de datos.
- Verificación de existencia del archivo.
- Separación de los parámetros.
- Comprobación de los valores.
- División en conjuntos de entrenamiento y prueba.
- Creación del modelo: Por medio de la clase o función *SVC* se elabora el modelo, en donde se agregan los cuatro hiperparámetros hallados previamente.
- Entrenamiento del modelo.
- Predicciones del modelo.
- Evaluación del modelo.
- Excepciones y errores.

En esta ocasión solo se explicó una de las 10 secciones, debido a que cada una de estas ya fue explicada en la sección 7.5.1 y cumplen con la misma funcionalidad.

Una vez se conocen las secciones que componen a la creación del modelo y se crea este mismo, se procede a observar los resultados obtenidos.

Capítulo 8

Resultados

8.1. Random Forest

Con el modelo ya creado con cada uno de los hiperparámetros optimizados se pone en funcionamiento el modelo de ML. A continuación, se mostrarán los resultados obtenidos y se explicarán cada uno de ellos:

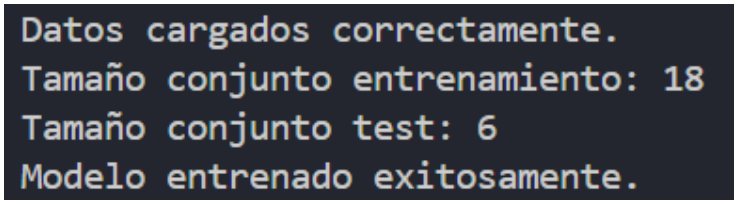


Figura 8.1: Resultado de la carga y conjunto de datos del modelo RF.

En la figura 8.1 se puede observar que los datos del archivo csv se cargaron de manera correcta al DataFrame, el conjunto de entrenamiento tomado es de 18 sujetos de prueba y el conjunto de prueba es de 6 sujetos, esto permite que el modelo haya sido entrenado correctamente.

```
Predicción del modelo: [1 1 1 1 0 1]
Resultados esperados de la predicción:
23    1
6     1
13    1
3     1
10    0
20    0
Name: Enfermo, dtype: int64
Exactitud del modelo: 0.83
```

Figura 8.2: Predicción del modelo, resultados esperados y exactitud de este mismo.

En la figura 8.2, se pueden observar las predicciones realizadas por el modelo, los resultados esperados de la predicción y la exactitud del modelo. Los resultados esperados son los valores reales del diagnóstico de cada paciente, se debe tener en cuenta que aquellos pacientes que padecen del SM están identificados con el número 1 y aquellos que no sufren de esta enfermedad están identificados con el número 0, haciendo alusión al estado alto y bajo de los sistemas digitales. La predicción del modelo falla en un valor haciendo así que el modelo tenga una exactitud del 83 %, esto se debe a que la cantidad de sujetos que se tiene para entrenar la ML no es muy grande y además la diferencia entre los datos de cada uno de los parámetros de entrenamiento, en muchos casos, no representa una diferencia significativa, por lo cual el modelo al realizar la clasificación y la toma de decisiones presentará fallas.

A pesar de que el modelo RF se considera preciso y robusto, se pueden presentar errores en la toma de decisiones al momento de realizar clasificaciones, más aún cuando se tienen valores tan similares como es el caso de los datos extraídos de cada uno de los sujetos de estudio, por lo cual, este también es otro factor que incide en el correcto diagnóstico.

El hiperparámetro de *random_state* incide directamente en la exactitud del modelo, esto se debe a que es el encargado de decidir cuáles son los datos de entrenamiento y de prueba. Como se mencionó anteriormente, ciertos datos presentan una similitud independientemente del grupo al cual pertenezca, haciendo más difícil la tarea de clasificación para el modelo, por lo que dependiendo con que datos se entrene y se pruebe el modelo se tendrá una exactitud mayor. Esto se pudo observar en la figura 7.24, en donde para cada valor de este hiperparámetro se tiene un valor de exactitud diferente, esto demuestra que no todos los datos son óptimos para entrenamiento o prueba. Haciendo un promedio de la

exactitud del modelo, por medio de la variación de 0 hasta 100 del valor de *random_state*, se obtiene lo siguiente:

```
Promedio de todas las exactitudes obtenidas:
Promedio de exactitudes: 0.44
```

Figura 8.3: Promedio de las exactitudes del modelo dependiendo del random_state desde 0 hasta 100.

Como se observa en la figura 8.3, el modelo en cuestión, independientemente de los valores escogidos para entrenar y realizar las pruebas presenta una exactitud promedio del 44 %, en donde este valor es demasiado bajo para una predicción de una enfermedad y donde también se demuestra la hipótesis antes planteada de que los datos de entrenamiento y prueba afectan directamente a la precisión y/o exactitud del modelo/diagnóstico.

Como se mencionó anteriormente se realizaron 100 pruebas en para obtener la exactitud promedio del modelo RF, esto con lleva a una carga significativa tanto de tiempo como de recursos computacionales. En donde, en la figura 8.4, se puede observar los recursos utilizados por el modelo en cuestión:

```
--- Resumen final de recursos utilizados ---
Tiempo total de ejecución: 183.27 segundos
Memoria usada total: 1.43 MB
Memoria máxima utilizada: 1.86 MB
Memoria inicial del proceso: 142.23 MB
Memoria final del proceso: 149.71 MB
Uso inicial de CPU: 26.70%
Uso final de CPU: 32.10%
Hilos iniciales: 22
Hilos finales: 16
Lecturas en disco: 10128
Escrituras en disco: 4551
```

Figura 8.4: Recursos computacionales utilizados por el método RF.

Como se observa en la figura 8.4, se hizo el análisis del tiempo total de ejecución, el cual es un poco elevado debido a que se deben generar 150 árboles cada vez que se modifica el

valor de *random_state*, es decir, se genera 15000 en total. Por lo cual, esto ocupa un espacio de memoria 7.48 MB, el uso de la memoria de CPU durante el proceso es de 5.4 % lo cual es muy bajo y no afectaría el rendimiento del equipo. El cambio de uso de los hilos iniciales y los finales se puede deber al comportamiento de gestión de hilos o procesos de Python, por lo cual inicialmente se tienen 22 hilos para la ejecución de la ML y luego se reducen a 16 hilos, donde se muestra que a partir de determinado momento ya no se requiere el uso de la cantidad de hilos iniciales. Las lecturas de disco hace referencia a la cantidad de veces que la ML realizó una lectura de un archivo en el disco duro, en este caso se refiere a la cantidad de veces que leyó el archivo csv donde se tienen todos los parámetros hallados en 7.4, este número puede parecer alto, sin embargo, es un valor normal debido a que el archivo tiene más de 10000 datos los cuales tiene que leer 100 veces para realizar el proceso de las predicciones. Por último, se tienen las escrituras en disco, las cuales están directamente relacionadas a las veces que se guardaron datos en el disco duro en cada uno de los procesos realizados.

8.2. Support Vector Machine

Con el modelo ya creado con cada uno de los hiperparámetros optimizados, se pone en funcionamiento el modelo del ML. A continuación, se mostrarán los resultados obtenidos y se explicarán cada uno de ellos:

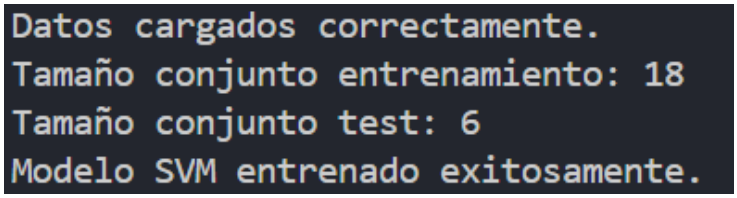


Figura 8.5: Resultado de la carga y conjunto de datos del modelo SVM.

En la figura 8.5 se puede observar que los datos del archivo csv se cargaron de manera correcta al DataFrame, el conjunto de entrenamiento tomado es de 18 sujetos de prueba y el conjunto de prueba es de 6 sujetos, esto permite que el modelo haya sido entrenado correctamente.

```
Predicción del modelo: [0 0 1 0 1 1]
Resultados esperados de la predicción:
20    0
2     0
17    1
18    0
19    1
23    1
Name: Enfermo, dtype: int64
Exactitud del modelo: 1.00
```

Figura 8.6: Predicción del modelo, resultados esperados y exactitud de este mismo.

En la figura 8.6 se puede observar las predicciones que realizó el modelo con base al entrenamiento que se llevó a cabo previamente y los resultados esperados, que son el conjunto de datos que identifican a cada sujeto, esté enfermo o no. Recordemos que aquellos sujetos de estudio que se encuentren identificados con el número 1 son los que padecen la enfermedad del SM y aquellos sujetos que no sufren de dicha enfermedad son identificados con el número 0. Para esta predicción, en donde el valor de *random_state* es de 5, el modelo no presenta ningún error al momento de dar los diagnósticos correspondientes, sin embargo, esto no quiere decir que el modelo sea 100 % efectivo. Se podrá observar la exactitud total del modelo en la figura 8.7.

Como se mencionó en los resultados de la sección 7.5.1, el hiperparámetro de *random_state* incide directamente en la exactitud del modelo. Para este modelo de SVM dicha directriz no cambia, esto se pudo observar en la figura 7.26 en donde se tienen diferentes valores de *random_state* con diferentes exactitudes. Realizando un promedio de la exactitud del modelo, por medio de la variación de 0 hasta 100 de dicho hiperparámetro, se obtiene lo siguiente:

```
Probando con random_state=100
Exactitud: 0.67
```

Figura 8.7: Promedio de las exactitudes del modelo dependiendo del random_state desde 0 hasta 100.

Como se observa en la figura 8.7, se vuelve a presentar la disminución de la exactitud dependiendo de que valores se tomen para el entrenamiento y la prueba en el modelo, en donde para el SVM se tiene un 66 %. Confirmando una vez más la hipótesis planteada en la sección 7.5.1.

Como se mencionó anteriormente, se realizaron 100 pruebas en para obtener la exactitud promedio del modelo SVM, esto con lleva a una carga significativa tanto de tiempo como de recursos computacionales. En donde, en la figura 8.8, se puede observar los recursos utilizados por el modelo en cuestión:

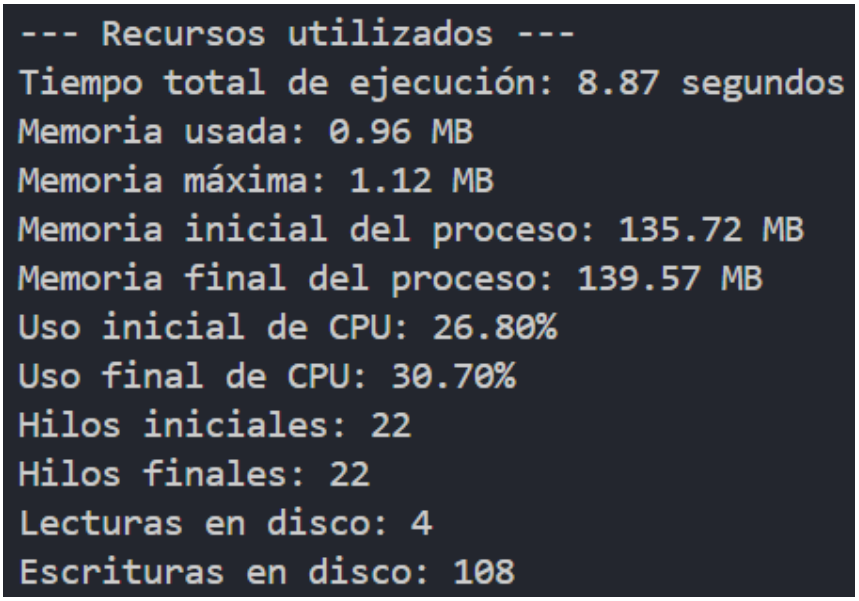


Figura 8.8: Recursos computacionales utilizados por el método SVM.

Como se observa en la figura 8.8, se hizo el análisis del tiempo total de ejecución, el cual es bastante bajo, esto se relaciona con que el modelo solo debe leer el archivo csv de los parámetros una vez y puede ejecutar el resto de procesos con esta información almacenada. Por lo cual, esto ocupa un espacio de memoria 3.85 MB, el uso de la memoria de CPU durante el proceso es de 3.9 % lo cual es muy bajo y no afectaría el rendimiento del equipo. Esto está directamente relacionado con la baja lectura de los parámetros, donde con tal solo 4 veces que se realice este proceso se puede realizar las tareas correspondientes. El uso de los hilos no varía, indicando que durante este proceso fueron necesarios los 22 hilos para llevar a cabo este proceso. Las lecturas de disco hace referencia a la cantidad de veces que la ML realizó una lectura de un archivo en el disco duro, en este caso se refiere a la cantidad de veces que leyó el archivo csv donde se tienen todos los parámetros hallados

en 7.4, al tener un valor de 4 significa que con esta cantidad de lecturas se ejecutó las predicciones. Por último, se tienen las escrituras en disco, las cuales están directamente relacionadas con las veces que se guardaron datos en el disco duro en cada uno de los procesos realizados.

8.3. Comparación de los modelos RF vs SVM

En esta sección se presenta un análisis comparativo de los resultados obtenidos por los modelos de machine learning implementados: Random Forest y Support Vector Machine. El propósito de esta comparación es evaluar el desempeño de ambos modelos en la detección del síndrome metabólico a partir de señales electrocardiográficas.

El análisis se centra en dos aspectos clave: los resultados generales obtenidos por cada modelo, donde se abarca su exactitud en la predicción por medio de las señales ECG y los recursos computacionales consumidos durante la ejecución de cada modelo.

8.3.1. Resultados generales

Los modelos de ML tienen como objetivo principal el realizar la predicción de las personas que puedan o padezcan del SM por medio de las señales del corazón ECG. Para llevar a cabo este proceso se debe tener en cuenta la exactitud final que presentan cada uno de ellos, siendo este un factor determinante al momento de decidir cuál es el modelo de mayor confianza.

En cada uno de los dos modelos se realizaron 100 pruebas por medio de la variación del hiperparámetro *random_state* como se mencionó ha mencionado en la sección 7.5, por medio de estas pruebas se puede determinar la exactitud promedio que tienen los modelos al momento de realizar las predicciones. Estos valores se observan en la siguiente tabla 8.1:

Modelo	**Exactitud promedio**
RF	0.44
SVM	0.67

Tabla 8.1: Exactitud promedio de los dos modelos de ML.

Con lo observando en la tabla 8.1 se puede determinar que la ML implementada por medio de SVM tiene una exactitud promedio más alta, con lo cual es la más confiable al momento de tomar en cuenta sus predicciones. Esto se puede ver representado en la siguiente figura 8.9:

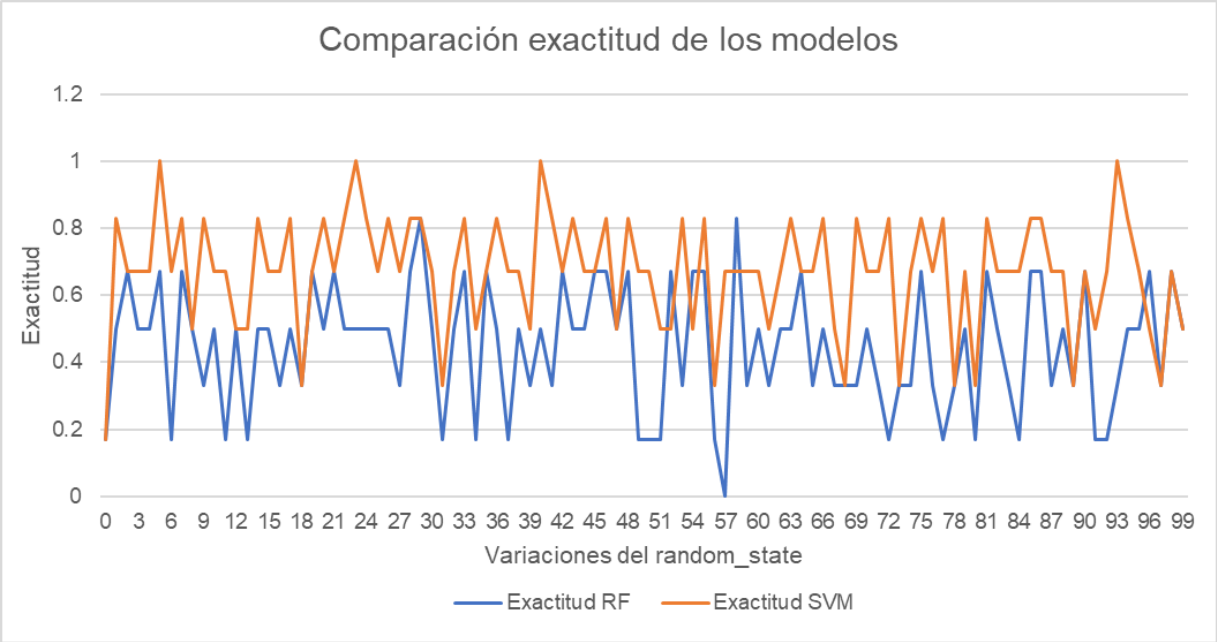


Figura 8.9: Comparación de las exactitudes de los modelos.

En la figura 8.9 se observa que el modelo SVM presenta un comportamiento más constante en sus valores de exactitud y de mayor porcentaje, en comparación con el modelo RF. Además, tiene valores más altos de exactitud, que se repiten en varias ocasiones en el modelo SVM. Demostrando así porque este modelo posee una exactitud promedio mayor (con un 67 %) que la del modelo RF (con un 44 %).

Si se habla desde las bases teóricas, se debe poner el foco de atención en el número de predictores. Para llevar a cabo el proyecto se hace uso de 680 predictores, los cuales toman distintos valores y por medio de estos se entrena el modelo y se realiza la predicción. Al momento de tener esta gran cantidad de predictores y no existir una diferencia significativa entre muchos de ellos, pueden llegar a no ser los mejores entrenadores para los modelos, generando fallas en el aprendizaje. Sin embargo, es allí donde el modelo SVM saca ventaja sobre el modelo RF, mientras que este último modelo requiere de todos los predictores para realizar su aprendizaje, clasificaciones y/o predicciones, el modelo SVM al tener una franja de aceptación y decisión de cuáles predictores son útiles para realizar el proceso de clasificación, puede eliminar todo aquel valor que no cumpla con las características necesarias que lo ayuden a realizar una correcta predicción, reduciendo así en gran cantidad el número de predictores a utilizar, permitiéndole ser más exacto en los diagnósticos arrojados.

Con lo anterior descrito, se puede hablar de que el entrenamiento del modelo SVM es más preciso, ya que, sin importar los datos que se le ingresen para este proceso, tiene valores más altos y constantes de exactitud en comparación con el modelo RF.

8.3.2. Recursos computacionales consumidos

Los recursos computacionales son otro de los factores determinantes al momento de seleccionar el mejor modelo posible. Como se mostró en las secciones 8.1 y 8.2, se tienen los siguientes consumo de recursos computacionales de cada uno de los modelos:

Consumo de recursos computacionales		
Recurso	**Random Forest**	**Support Vector Machine**
Tiempo de ejecución	183.27 s	8.87 s
Memoria usada total	1.43 MB	0.96 MB
Memoria máxima usada	1.86 MB	1.12 MB
Memoria inicial del proceso	142.23 MB	135.72 MB
Memoria final del proceso	149.71 MB	139.57 MB
Uso inicial de CPU	26.70 %	26.08 %
Uso final de CPU	32.10 %	30.70 %
Lecturas en disco	10128	4
Escrituras en disco	4551	108

Tabla 8.2: Recursos computacionales consumidos por cada uno de los métodos.

Como se observa en la tabla 8.2, el modelo que más recursos consumo en general es el de RF, tanto en tiempo como en todos los procesos relacionados con la memoria, disco duro y de CPU. Esto se debe a que este modelo debe estar en constante realización de decisiones y clasificaciones por medio de cada uno de los árboles que se crean, además de esto tiene en cuenta todos los 680 predictores para llevar a cabo el diagnóstico. Si se realiza un análisis detallado de los parámetros se puede decir que:

- El tiempo de ejecución del modelo RF es mayor al del modelo SVM debido a que toma todos los predictores para el entrenamiento y diagnóstico final, debe leerlos 100 veces. Además, debe crear 150 árboles de decisión por cada proceso que realicé, generando así un total de 15000 árboles. En cambio, el modelo SVM solo tiene en cuenta los predictores que no incumplan las normas de frontera establecidas, reduciendo así la gran cantidad de predictores a analizar.
- En el ámbito de los recursos relacionados con la memoria, se observa que el modelo RF hace un uso un poco más grande de este recurso, esto se debe a la gran cantidad de procesos que debe elaborar durante su ejecución.

- El porcentaje del uso inicial de la CPU tiene un valor cercano para los dos modelos, esto suceda porque se inicia con la lectura de los predictores y se guardan en una variable determinada. Inicialmente, se tienen procesos parecidos para cada modelo, es a medida que va a avanzando el entrenamiento y las predicciones donde se presenta un uso mayor por parte del modelo RF, ya que, como se explicó anteriormente, debe realizar una cantidad considerable de árboles de decisión.

- Las lecturas en disco son mayores para el modelo RF debido a que este debe estar en constante revisión de los predictores y sus valores para así realizar tanto su entrenamiento como su diagnóstico. En cambio, el modelo SVM solo requiere de unas cuantas lecturas para realizar este mismo proceso, no se puede olvidar que el SVM reduce la cantidad de predictores, permitiéndole así llevar a cabo el diagnóstico de una manera más sencilla y rápida.

- Las escrituras en disco del modelo RF presentan tanta diferencia con las del modelo SVM, debido a que se generan 15000 árboles y se guardan todos estos datos.

Por lo que el modelo que menos recursos computacionales consume es el SVM. Convirtiéndolo, para esta aplicación, en el modelo más exacto, confiable y óptimo, viéndolo desde un ámbito de diagnóstico y computacional.

Capítulo 9

Conclusiones

- El proyecto propone un método no invasivo basado en el análisis de señales electrocardiográficas utilizando técnicas de machine learning. Esto representa un avance significativo en comparación con los métodos tradicionales, que son invasivos, y tediosos; este enfoque mejora la experiencia de los pacientes al reducir los tiempos de diagnóstico, además permite el acceso a una herramienta de detección temprana lo cual beneficia a diversas poblaciones.

- Se demuestra como la ingeniería electrónica es capaz de generar soluciones prácticas y efectivas para problemas médicos complejos, resaltando la relevancia de la aplicación de tecnología avanzada en el ámbito de la salud.

- El preprocesamiento de datos toma un papel de gran importancia en este tipo de aplicación, debido a que le permite a la máquina poder entender de forma óptima los datos que se le están suministrando. Es por ello que es de vital importancia realizar este proceso antes de comenzar con la elaboración de cualquier proceso de aprendizaje automático.

- La aplicación de conceptos electrónicos como lo son el filtrado de las señales, en este caso por medio de filtros digitales, permite una mejor comprensión de las señales ECG, facilita el manejo de estas mismas y se pueden realizar los procesos posteriores de forma óptima, sin preocuparse por las alteraciones en los valores que pueden ocasionar agentes externos no deseados como lo es el ruido.

- Las bases de datos contaban con dos grupos de estudio, el diagnosticado con síndrome metabólico y el de control, estos tenían 14 pacientes de estudio y 10 pacientes de estudio respectivamente. Al ser una base de datos de 24 pacientes, se presentan limitaciones al momento de realizar el entrenamiento de cada uno de los modelos, debido a que para procesos tan robustos esta cantidad de datos resulta ser un poco

limitada y no se logra realizar el proceso de entrenamiento correctamente, impidiendo poder generar patrones más confiables. Además, a esto se le suma el hecho de que estos datos no presentan una gran diferencia numérica entre muchos de ellos, llegando a general confusiones al momento de tomar decisiones o clasificaciones en cada uno de los modelos.

- Los resultados obtenidos al aplicar modelos de machine learning, como Random Forest y Support Vector Machine, demostraron su capacidad para clasificar con exactitud las señales ECG y detectar patrones característicos del SM. Aunque ambos métodos fueron efectivos, se realizaron análisis comparativos que permitieron identificar cuál es más adecuado en términos de exactitud, tiempo de procesamiento y consumo de recursos computacionales. El modelo de SVM obtuvo mejores resultados, con un 67 % de exactitud promedio, debido a que su metodología de trabajo se basa en la reducción de observaciones no lineales y la transformación de espacios vectoriales, por lo cual selecciona únicamente los predictores realmente importantes y/o determinantes para llevar a cabo el entrenamiento y diagnóstico. Por otra parte, el modelo RF presenta el menor rendimiento tanto en la exactitud y confiabilidad como en el consumo de recursos computacionales, esto se debe a su forma de trabajo, en donde este debe realizar un número de procesos considerable en comparación del modelo SVM. Al momento de realizar la generación de cada árbol de decisión, siempre toma en cuenta todos y cada uno de los predictores presentes, comprometiendo su exactitud y aumentando los recursos computacionales consumidos.

- Este proyecto sienta una base sólida para estudios futuros en los cuales se podrían integrar otras modalidades de señales biométricas y así enriquecer el análisis de estas mismas. Asimismo, la colaboración con profesionales en el campo de la salud permitiría perfeccionar los modelos y así fortalecer la aceptación de estas técnicas en el ámbito clínico.

- Durante esta investigación se presentaron limitantes como lo son la disponibilidad de sujetos de prueba al contar con una base de datos limitada, se presenta reducción de los niveles de exactitud y no se realiza el entrenamiento más óptimo, evitando que cada uno de los modelos aprendan de la manera más efectiva. Además, no se cuenta con la validación médica correspondiente para garantizar la aplicabilidad clínica del método.

- Este trabajo representa un cambio en el paradigma del diagnóstico de enfermedades cardiacas, más específicamente en el diagnóstico del síndrome metabólico, al proponer un método, innovador, eficiente y accesible.

Referencias

[1] M. P. Martínez, I. D. Vergara, K. Q. Molano, M. M. Perez y A. P. Ospina, "Síndrome Metabólico en Adultos: Revisión Narrativa de la Literatura," *iMedPub Journals*, 2021. dirección: https://www.archivosdemedicina.com/medicina-de-familia/siacutendrome-metaboacutelico-en-adultos-revisioacuten-narrativa-de-la-literatura.pdf.

[2] A. Muchnik. "La OMS revela las principales causas de muerte y discapacidad en el mundo: 2000-2019." (2020), dirección: https://www.who.int/es/news/item/09-12-2020-who-reveals-leading-causes-of-death-and-disability-worldwide-2000-2019#:~:text=Las%20cardiopat%C3%ADas,%20la%20diabetes,%20los,2019,%20en%20comparaci%C3%B3n%20con%202000..

[3] R. A. Española. "prevalencia." (2014), dirección: https://dle.rae.es/prevalencia.

[4] M. C. Fragozo-Ramos, "Síndrome metabólico: revisión de la literatura," *Biblioteca virtual en salud*, 2022. dirección: https://docs.bvsalud.org/biblioref/2022/06/1370951/sindrome-metabolico.pdf.

[5] DANE, "Estadísticas Vitales (EEVV) Defunciones fetales y no fetales 3er trimestre 2023pr, año corrido 2023pr, cifras definitivas 2022," Departamento Administrativo Nacional de Estadística, inf. téc., 2023. dirección: https://www.dane.gov.co/files/operaciones/EEVV/bol-EEVV-Defunciones-IIItrim2023.pdf.

[6] S. RUGELES, J. F. CASTRO y Á. J. BORRERO, "Errores en la atención en salud: estudio piloto para el diseño de procesos más seguros en el Hospital Universitario San Ignacio," *Revista Colombiana de Cirugía*, 2004. dirección: http://www.scielo.org.co/pdf/rcci/v19n2/v19n2a8.pdf.

[7] E. M. A.D.A.M. "Prueba de tolerancia a la glucosa - que no están embarazadas." (2023), dirección: https://medlineplus.gov/spanish/ency/article/003466.htm#:~:text=Es%20una%20prueba%20de%20laboratorio,para%20el%20diagn%C3%B3stico%20de%20diabetes.

[8] N. Tentolouris, G. Argyrakopoulou y N. Katsilambros, "Perturbed Autonomic Nervous System Function in Metabolic Syndrome," *NeuroMolecular Medicine*, 2008. dirección: https://link.springer.com/article/10.1007/s12017-008-8022-5#:~:text=Subjects%20with%20metabolic%20syndrome%20have,%2C%20adipose%20tissue%2C%20and%20muscles..

[9] Mayo Clinic, *Síndrome metabólico - Síntomas y causas*, https://www.mayoclinic.org/es-es/diseases-conditions/metabolic-syndrome/symptoms-causes/syc-20351916, Recuperado de Mayo Clinic, jun. de 2019.

[10] C. Carvajal Carvajal, "Síndrome metabólico: definiciones, epidemiología, etiología, componentes y tratamiento," *Medicina Legal de Costa Rica*, vol. 34, n.º 1, págs. 175-193, 2017, Recuperado el 12 de abril de 2024. dirección: http://www.scielo.sa.cr/scielo.php?script=sci_arttext&pid=S1409-00152017000100175&lng=en&tlng=es.

[11] A. Rodríguez, H. Delgado-Cohen, J. Reviriego y M. Serrano-Ríos, "Risk factors associated with metabolic syndrome in type 2 diabetes patients according to World Health Organization, Third Report National Cholesterol Education Program, and International Diabetes Federation definitions," *Diabetes, Metabolic Syndrome and Obesity: Targets and Therapy*, vol. 4, págs. 1-4, 2011.

[12] E. Kassi, P. Pervanidou, G. Kaltsas y G. Chrousos, "Metabolic syndrome: definitions and controversies," *BMC Medicine*, vol. 9, págs. 1-13, 2011.

[13] S. S. Anand, M. A. Tarnopolsky, S. Rashid, K. Schulze, D. Desai, A. Mente et al., "Adipocyte hypertrophy, fatty liver, and metabolic risk factors in South Asians: The Molecular Study of Health and Risk Ethnic Groups (mol.SHARE)," *PLOS One*, vol. 6, n.º 7, págs. 1-8, 2011.

[14] M. Chandalia, P. Lin, T. Seenivasan et al., "Insulin resistance and body fat distribution in South Asian men compared to Caucasian men," *PLOS One*, vol. 2, n.º 8, págs. 1-7, 2007.

[15] J. Tuomilehto, J. Lindstrom, J. G. Eriksson, T. T. Valle, H. Hamalainen, P. Ilanne-Parikka et al., "Prevention of type 2 diabetes mellitus by changes in lifestyle among subjects with impaired glucose tolerance," *New England Journal of Medicine*, vol. 344, n.º 18, págs. 1343-1350, 2001.

[16] X. R. Pan, G. W. Li, Y. H. Hu, J. X. Wang, W. Y. Yang, Z. X. An et al., "Effects of diet and exercise in preventing NIDDM in people with impaired glucose tolerance. The Da Qing IGT and Diabetes Study," *Diabetes Care*, vol. 20, n.º 4, págs. 537-544, 1997.

[17] American Diabetes Association, "Standards of medical care in diabetes–2012," *Diabetes Care*, vol. 35, n.º Suppl 1, S11-S63, 2012.

[18] National Institutes of Health, "Clinical Guidelines on the Identification, Evaluation, and Treatment of Overweight and Obesity in Adults–The Evidence Report," *Obesity Research*, vol. 6, n.º Suppl 2, 51S-209S, 1998.

[19] D. B. Sacks, D. E. Bruns, D. E. Goldstein, N. K. Maclaren, J. M. McDonald y M. Parrott, "Guidelines and recommendations for laboratory analysis in the diagnosis and management of diabetes mellitus," *Clinical Chemistry*, vol. 48, n.º 3, págs. 436-472, 2002.

[20] Empendium, *Prueba de tolerancia oral a la glucosa*, https://empendium.com/manualmibe/tratado/chapter/B76.IV.L.2.9.1, Fuente, n.d.

[21] Cascino, Thomas and Shea, Michael J., *Electrocardiografía*, https://www.msdmanuals.com/es-co/professional/trastornos-cardiovasculares/pruebas-y-procedimientos-cardiovasculares/electrocardiografa, Recuperado de Manual MSD, jul. de 2021.

[22] T. Cascino y M. J. Sea. "Electrocardiografía." (2021), dirección: https://www.msdmanuals.com/es/professional/trastornos-cardiovasculares/pruebas-y-procedimientos-cardiovasculares/electrocardiograf%C3%ADa.

[23] F. Miyara, *FILTROS ACTIVOS*. Universidad Nacional de Rosario, 2004. dirección: https://avcontractor.com.mx/pdf/FiltrosActivos_Ing.FedericoMiraya.pdf.

[24] G. M. Smith. "¿Qué es el procesamiento de señales?" (2023), dirección: https://dewesoft.com/es/blog/que-es-procesamiento-de-senal.

[25] J. M. M. de la Rosa. "FUNDAMENTOS TEÓRICOS: FILTROS." (), dirección: https://dewesoft.com/es/blog/que-es-procesamiento-de-senal.

[26] J. G. Proakis y D. G. Manolakis, *Tratamiento digital de señales*. PEARSON, 2007.

[27] Anónimo. "Filtros activos filtro pasa bajas Butterworth." (2018), dirección: https://gc.scalahed.com/recursos/files/r145r/w868w/U3_liga3.html.

[28] A. M. Cuadrado, "UTILIZACIÓN DEL MACHINE LEARNING EN LA INDUSTRIA 4.0," Tesis de mtría., Universidad de Valladolid, 2019. dirección: https://uvadoc.uva.es/handle/10324/37908.

[29] A. F. Khatiboun, "Machine Learning en Ciberseguridad," Tesis de mtría., Universitat Oberta de Catalunya, 2019. dirección: https://openaccess.uoc.edu/bitstream/10609/97546/7/afernandezkhTFM0619memoria.pdf.

[30] F. de Milagro Paullo Montes, "Comparación de modelos de Machine Learning para determinar qué modelo se aproxima más a una asignación manual en ServiceDesk," Tesis de mtría., Universidad Nacional Mayor de San Marcos, 2023. dirección: https://cybertesis.unmsm.edu.pe/bitstream/handle/20.500.12672/21853/Paullo_mf.pdf?sequence=1&isAllowed=y.

[31] J. L. F. Genaro, "MODELOS DE MACHINE LEARNING PARA LA CIENCIA DE DATOS," Tesis de mtría., Universidad de Salamanca, 2023. dirección: https://gredos.usal.es/bitstream/handle/10366/157171/TFG%20Juan%20Luis%20Fern%C3%A1ndez%20Genaro.pdf?sequence=1&isAllowed=y.

[32] F. Khan. "¿Qué es el preprocesamiento de datos? Definición, importancia y pasos." (2024), dirección: https://www.astera.com/es/type/blog/data-preprocessing/#:~:text=Implica%20limpiar%2C%20transformar%20y%20organizar,efectiva%20y%20obtener%20informaci%C3%B3n%20significativa..

[33] C. E. F. Tapia y K. L. F. Cevallos, "PRUEBAS PARA COMPROBAR LA NORMALIDAD DE DATOS EN PROCESOS PRODUCTIVOS: ANDERSONDARLING, RYAN-JOINER, SHAPIRO-WILK Y KOLMOGÓROV-SMIRNOV," *SOCIETAS*, 2021. dirección: http://portal.amelica.org/ameli/jatsRepo/341/3412237018/index.html.

[34] Bscan. "Prueba de Kolmogórov-Smirnov." (2023), dirección: https://es.wikipedia.org/wiki/Prueba_de_Kolmog%C3%B3rov-Smirnov#/media/Archivo:KS_Example.png.

[35] M. R. Saldaña, "Contraste de Hipótesis Comparación de dos medias independientes mediante pruebas no paramétricas: Prueba U de Mann-Whitney," *Revista Enfermería del Trabajo*, 2013. dirección: https://dialnet.unirioja.es/servlet/articulo?codigo=4327652.

[36] M. Narvaez. "Prueba U de Mann-Whitney: Qué es y cómo funciona." (2023), dirección: https://www.questionpro.com/blog/es/prueba-u-de-mann-whitney/#:~:text=La%20prueba%20U%20de%20Mann-Whitney%20realiza%20una%20comparaci%C3%B3n%20estad%C3%ADstica,tanto%2C%20de%20la%20misma%20poblaci%C3%B3n..

[37] S. S. Soflaei, N. Varasteh, G. Pourali et al., "Association of Metabolic Syndrome and Ischemic Changes in ECG: Result from a Population-Based Study," *Research Square*, 2023. dirección: https://www.researchsquare.com/article/rs-3052616/v1.

[38] A. Farhadi, H. Emamat, R. Nemati et al., "The association between continuous metabolic syndrome score and its components with electrocardiographic abnormalities in community-dwelling older adults: the Bushehr elderly health (BEH) program," *BMC Cardiovascular Disorders*, 2024. dirección: https://bmccardiovascdisord.biomedcentral.com/articles/10.1186/s12872-024-03733-1.

[39] A. V. Mondragon, E. Llanes, O. Sison y P. Reganit, "Association of ECG Abnormalities with Metabolic Syndrome and Socio-demographic Factors in the Philippine LIFECARE Cohort," *Atherosclerosis and Thrombosis 21*, 2014. dirección: https://www.researchgate.net/publication/278329460_Association_of_ECG_Abnormalities_with_Metabolic_Syndrome_and_Socio-demographic_Factors_in_the_Philippine_LIFECARE_Cohort.

[40] T. W. Elffers, R. de Mutsert, H. J. Lamb et al., "Association of metabolic syndrome and electrocardiographic markers of subclinical cardiovascular disease," *Diabetology Metabolic Syndrome*, 2017. dirección: https://dmsjournal.biomedcentral.com/articles/10.1186/s13098-017-0238-9.

[41] W. Li, Y. Bai, K. Sun et al., "Patients with Metabolic Syndrome Have Prolonged Corrected QT Interval (QTc)," *Preventive Cardiology*, 2009. dirección: https://onlinelibrary.wiley.com/doi/10.1002/clc.20416.

[42] G. Perpiñan, E. Severeyn, M. A. Altuve y S. Wong, "VARIABILIDAD DE LA FRECUENCIA CARDÍACA DURANTE LA PRUEBA ORAL DE TOLERANCIA A LA GLUCOSA EN SUJETOS CON SÍNDROME METABÓLICO, SEDENTARIOS Y DEPORTISTAS," en *V Congreso de Venezolano de Bioingeniería*, 2015. dirección: https://www.researchgate.net/publication/281208663_VARIABILIDAD_DE_LA_FRECUENCIA_CARDIACA_DURANTE_LA_PRUEBA_ORAL_DE_TOLERANCIA_A_LA_GLUCOSA_EN_SUJETOS_CON_SINDROME_METABOLICO_SEDENTARIOS_Y_DEPORTISTAS.

[43] C. Lim, J.-Y. Kim e Y. Nam, "ECG Signal Analysis for Patient with Metabolic Syndrome based on 1D-Convolution Neural Network," en *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2020. dirección: https://ieeexplore.ieee.org/document/9457885.

[44] C. A. Ledezma, E. Severeyn, G. Perpiñán, M. Altuve y S. Wong, "A new on-line electrocardiographic records database and computer routines for data analysis," en *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2014. dirección: https://ieeexplore.ieee.org/document/6944189.

[45] CDC. "Cómo evaluar su peso." (2013), dirección: https://www.cdc.gov/healthyweight/spanish/assessing/index.html#:~:text=Si%20su%20IMC%20es%20menos,dentro%20del%20rango%20de%20sobrepeso..

[46] J. Berman. "Presión arterial." (2023), dirección: https://medlineplus.gov/spanish/ency/anatomyvideos/000013.htm#:~:text=Una%20presi%C3%B3n%20sist%C3%B3lica%20de%20120,como%20%22120%20sobre%2080%22..

[47] F. L. Jiménez. "Triglicéridos: ¿por qué son importantes?" (2022), dirección: https://www.mayoclinic.org/es/diseases-conditions/high-blood-cholesterol/in-depth/triglycerides/art-20048186.

[48] l. P. y. l. S. Instituto Nacional del Corazón. "Niveles de colesterol: Lo que usted debe saber." (2024), dirección: https://medlineplus.gov/spanish/cholesterollevelswhatyouneed.html.

[49] J. Berman. "Examen de VLDL." (2023), dirección: https://medlineplus.gov/spanish/ency/article/003494.htm#:~:text=El%20nivel%20de%20colesterol%20VLDL,2%20a%2030%20mg%2FdL..

[50] OMS. "La medida del perímetro abdominal es un indicador de enfermedad cardiovascular más fiable que el IMC." (2021), dirección: https://fundaciondelcorazon.com/prensa/notas-de-prensa/2264-medida-perimetro-abdominal-es-indicador-enfermedad-cardiovascular-mas-fiable-imc-.html#:~:text=La%20Organizaci%C3%B3n%20Mundial%20de%20la,valor%20es%20de%20102%20cent%C3%ADmetros..

[51] J. Berman. "Examen de sangre de alanina transaminasa (ALT)." (2023), dirección: https://medlineplus.gov/spanish/ency/article/003473.htm.

[52] N. J. Gonter. "Ácido úrico en la sangre." (2023), dirección: https://medlineplus.gov/spanish/ency/article/003476.htm#:~:text=Los%20valores%20normales%20est%C3%A1n%20entre,resultados%20espec%C3%ADficos%20de%20su%20examen..

[53] M. Clinic. "Análisis de nitrógeno ureico en sangre." (2023), dirección: https://www.mayoclinic.org/es/tests-procedures/blood-urea-nitrogen/about/pac-20384821#:~:text=En%20general%2C%20un%20nivel%20de,el%20laboratorio%20y%20tu%20edad..

[54] M. Lemos. "Índice de HOMA: qué significa, valores normales y qué indica." (2023), dirección: https://www.tuasaude.com/es/indice-de-homa/#:~:text=Valor%20normal%20del%20HOMA-IR,BETA%3A%20entre%20167%20y%20175..

[55] S. K. Dhaliwal. "Manejo de su glucemia." (2024), dirección: https://medlineplus.gov/spanish/ency/patientinstructions/000086.htm.

[56] D. T. C. at the University of California. "Información básica sobre la insulina." (2018), dirección: https://dtc.ucsf.edu/es/tipos-de-diabetes/diabetes-tipo-2/tratamiento-de-la-diabetes-tipo-2/medicamentos-y-terapias-2/prescripcion-de-insulina-para-diabetes-tipo-2/informacion-basica-sobre-la-insulina/#:~:text=La%20insulina%20natural%20(es%20decir,de%20las%20comidas%20y%20aperitivos..

[57] S. S. P. K. et al., "Heart Rate Variability Analysis for the Prediction of Metabolic Syndrome in Obese Children and Adolescents," *Indian Journal of Pediatrics*, vol. 87, n.º 2, págs. 92-97, 2020.

[58] A. N. P. A. et al., "Heart Rate Variability and Obesity Indices: Emphasis on the Response to the Glucose Tolerance Test," *Obesity Research & Clinical Practice*, vol. 12, n.º 6, págs. 526-533, 2018.

[59] H. E. et al., "Obesity and the Risk of Incident Diabetes Mellitus Among Medicare Beneficiaries," *Diabetes Care*, vol. 37, n.º 5, págs. 1222-1228, 2014.

[60] G. N. M. et al., "Obesity and Cardiovascular Risk Factors in Young Adults," *American Journal of Medicine*, vol. 129, n.º 5, págs. 564-570, 2016.

[61] H. A. E. et al., "The Metabolic Syndrome: A New Worldwide Definition," *The Lancet*, vol. 366, n.º 9491, págs. 1059-1062, 2005.

[62] L. N. et al., "The Association of Metabolic Syndrome and QRS—T Angle in US Adults," *NHANES III*, vol. 112, n.º 1, págs. 96-104, 2019.

[63] T. F. of the European Society of Cardiology, "Heart rate variability: Standards of measurement, physiological interpretation, and clinical use," *European Heart Journal*, vol. 17, págs. 354-381, 1996.

[64] Wellue. "Máquina de ECG de bolsillo de 6/12 derivaciones Máquina de ECG en el punto de atención en su bolsillo." (2024), dirección: https://es.getwellue.com/productos/M%C3%A1quina-de-ecg-de-bolsillo-de-12-derivaciones.

[65] A. C. Guyton y J. E. Hall, *Tratado de Fisiología Médica*, 12.ª ed. Philadelphia: Elsevier, 2011, Texto de referencia sobre fisiología cardíaca y ciclo cardíaco.

[66] G. J. M. Tomlinson et al., "Relationship between the PR Interval and Cardiovascular Risk Factors," *Journal of Cardiovascular Electrophysiology*, vol. 21, págs. 525-531, 2010.

[67] W. U. Arango, M. D. Ramírez y E. M. Durango, *Electrocardiografía y sus arritmias*. sociedad colombiana de cardiologia, 2010.

[68] M. Malik et al., "QT interval measurement and heart rate correction: to QTc or not to QTc?" *Journal of the American College of Cardiology*, vol. 36, págs. 746-752, 2000.

[69] A. L. Goldberger, Z. D. Goldberger y A. Shvilkin, *Goldberger's Clinical Electrocardiography: A Simplified Approach*, 9.ª ed. Philadelphia: Elsevier, 2020, Este texto incluye información detallada sobre el intervalo QTc corregido usando la fórmula de Bazett, y proporciona los valores normales para adultos.

[70] D. de Noronha, S. Maurício e I. Rodrigues, "Long QT Syndrome and Anaesthesia," *World Federation of Societies of Anasesthesiologists*, 2021. dirección: `https://resources.wfsahq.org/atotw/long-qt-syndrome-and-anaesthesia/`.

[71] H. Parrales. "Segmento ST." (2023), dirección: `https://cerebromedico.com/electrocardiograma/`.

[72] G. Reyes del Paso, S. Garrido, Á. Pulgar, M. Martín-Vázquez y S. Duschek, "Autonomic cardiovascular control and responses to experimental pain stimulation in fibromyalgia syndrome," *Journal of Psychosomatic Research*, vol. 75, n.º 3, págs. 275-283, 2013.

[73] M. Altuve, "Variabilidad de la frecuencia cardíaca: Técnicas temporales, frecuenciales y no lineales," en *VI Congreso Venezolano de Bioingeniería (BIOVEN)*, 2017, págs. 23-26.

[74] T. Benichou, B. Pereira, M. Mermillod et al., "Heart rate variability in type 2 diabetes mellitus: A systematic review and meta-analysis," *PLoS One*, vol. 13, n.º 4, e0195166, 2018.

[75] F. Dutheil, T. Benichou, H. Perera et al., "Monitoring heart rate variability in patients with diabetes mellitus: A systematic review and meta-analysis," *PLoS One*, vol. 13, n.º 8, e0195166, 2018.

[76] J. A. Rodrigo. "Random Forest con Python." (2020), dirección: `https://cienciadedatos.net/documentos/py06_machine_learning_python_scikitlearn`.

[77] J. A. Rodrigo. "Random Forest con Python." (2020), dirección: `https://cienciadedatos.net/documentos/py08_random_forest_python#Importancia-de-predictores`.

[78] J. A. Rodrigo. "Máquinas de Vector Soporte (SVM) con Python." (2020), dirección: `https://cienciadedatos.net/documentos/py24-svm-pythons`.

Anexos

9.1. Código en Python del modelo ML de Random Forest

```
1
2  import pandas as pd
3  from sklearn.model_selection import train_test_split
4  from sklearn.ensemble import RandomForestClassifier
5  from sklearn.metrics import accuracy_score
6  import numpy as np
7  import tracemalloc
8  import time
9  import psutil
10 import os
11
12 # Funcion para entrenar y evaluar el modelo
13 def train_random_forest(csv_file, Enfermo, random_state=None):
14     try:
15         # Cargar los datos desde el archivo
16         data = pd.read_csv(csv_file)
17
18         # Verificar si el archivo contiene la columna objetivo
19         if Enfermo not in data.columns:
20             raise ValueError(f"La columna objetivo '{Enfermo}' no esta
                   en los datos.")
21
22         # Separar caracteristicas (X) y etiquetas (y)
23         X = data.drop(columns=[Enfermo])
24         y = data[Enfermo]
25
26         # Asegurar que no haya valores nulos
27         if X.isnull().sum().any() or y.isnull().sum().any():
28             raise ValueError("Los datos contienen valores nulos. Por
                   favor, limpialos antes de continuar.")
```

```
29
30          # Dividir los datos en conjuntos de entrenamiento y prueba
31          X_train, X_test, y_train, y_test = train_test_split(X, y,
                test_size=0.25, random_state=random_state)
32
33          # Crear el modelo Random Forest
34          rf_model = RandomForestClassifier(n_estimators=150,
                max_depth=3, max_features=5, random_state=random_state)
35
36          # Entrenar el modelo
37          rf_model.fit(X_train, y_train)
38
39          # Realizar predicciones
40          y_pred = rf_model.predict(X_test)
41
42          # Evaluar el modelo
43          accuracy = accuracy_score(y_test, y_pred)
44
45          return accuracy, rf_model
46      except FileNotFoundError:
47          print(f"El archivo '{csv_file}' no fue encontrado.")
48      except ValueError as e:
49          print(f"Error en los datos: {e}")
50      except Exception as e:
51          print(f"Ocurrio un error: {e}")
52          return None, None
53
54  # Probar diferentes valores de random_state y medir recursos totales
55  def optimize_random_state(csv_file, Enfermo, max_trials=10):
56      # Iniciar medicion global de recursos
57      tracemalloc.start()
58      inicio_tiempo = time.time()
59      proceso = psutil.Process(os.getpid())
60      memoria_inicial = proceso.memory_info().rss / 1024 / 1024  #
            Memoria inicial en MB
61      cpu_inicial = psutil.cpu_percent(interval=None)  # Uso inicial de
            CPU
62      io_inicial = psutil.disk_io_counters()  # Operaciones de
            entrada/salida iniciales
63      hilos_iniciales = proceso.num_threads()  # Numero inicial de hilos
64
65      best_accuracy = 0
66      best_random_state = None
67      best_model = None
68      accuracies = []  # Lista para almacenar todas las exactitudes
69
70      for random_state in range(max_trials):
```

```
71          print(f"\nProbando random_state = {random_state}...")
72          accuracy, model = train_random_forest(csv_file, Enfermo,
                random_state=random_state)
73          if accuracy is not None:
74              accuracies.append(accuracy)
75              print(f"Exactitud obtenida: {accuracy:.2f}")
76
77          if accuracy > best_accuracy:
78              best_accuracy = accuracy
79              best_random_state = random_state
80              best_model = model
81
82      # Calcular estadisticas finales
83      fin_tiempo = time.time()
84      memoria_final = proceso.memory_info().rss / 1024 / 1024  # Memoria
            final en MB
85      uso_memoria = tracemalloc.get_traced_memory()
86      cpu_final = psutil.cpu_percent(interval=None)
87      io_final = psutil.disk_io_counters()
88      hilos_finales = proceso.num_threads()
89      temperatura = psutil.sensors_temperatures() if hasattr(psutil,
            "sensors_temperatures") else None
90      tracemalloc.stop()
91
92      tiempo_total = fin_tiempo - inicio_tiempo
93      memoria_usada = (uso_memoria[0] / 1024 / 1024)  # Memoria actual
            usada en MB
94      memoria_maxima = (uso_memoria[1] / 1024 / 1024)  # Memoria maxima
            en MB
95      mean_accuracy = np.mean(accuracies) if accuracies else 0
96      lecturas_dispositivo = io_final.read_count - io_inicial.read_count
97      escrituras_dispositivo = io_final.write_count -
            io_inicial.write_count
98
99      # Mostrar resumen final de recursos
100     print("\n--- Resumen final de recursos utilizados ---")
101     print(f"Tiempo total de ejecucion: {tiempo_total:.2f} segundos")
102     print(f"Memoria usada total: {memoria_usada:.2f} MB")
103     print(f"Memoria maxima utilizada: {memoria_maxima:.2f} MB")
104     print(f"Memoria inicial del proceso: {memoria_inicial:.2f} MB")
105     print(f"Memoria final del proceso: {memoria_final:.2f} MB")
106     print(f"Uso inicial de CPU: {cpu_inicial:.2f}%")
107     print(f"Uso final de CPU: {cpu_final:.2f}%")
108     print(f"Hilos iniciales: {hilos_iniciales}")
109     print(f"Hilos finales: {hilos_finales}")
110     print(f"Lecturas en disco: {lecturas_dispositivo}")
111     print(f"Escrituras en disco: {escrituras_dispositivo}")
```

```
    if temperatura:
        for name, entries in temperatura.items():
            for entry in entries:
                print(f"Temperatura {name} - {entry.label or
                    'desconocido'}: {entry.current} C ")

    # Mostrar estadisticas del modelo
    print("\n--- Resultados del modelo ---")
    print(f"Promedio de exactitudes: {mean_accuracy:.2f}")
    print(f"Mejor random_state: {best_random_state}")
    print(f"Mejor exactitud: {best_accuracy:.2f}")

    return best_random_state, best_accuracy, best_model, mean_accuracy

# Llamar a la funcion para optimizar el random_state
csv_file = "Ubicacion del archivo"
target_column = "Enfermo"
best_random_state, best_accuracy, best_model, mean_accuracy =
    optimize_random_state(csv_file, target_column, max_trials=100)
```

9.2. Código en Python del modelo ML de Support Vector Machine

```
    import pandas as pd
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report
import numpy as np
import tracemalloc
import time
import psutil
import os

# Funcion principal
def find_best_svm_model(csv_file, Enfermo, random_state_range):
    try:
        # Iniciar medicion de recursos
        tracemalloc.start()
        inicio_tiempo = time.time()
        proceso = psutil.Process(os.getpid())
        memoria_inicial = proceso.memory_info().rss / 1024 / 1024  #
            Memoria inicial en MB
        cpu_inicial = psutil.cpu_percent(interval=None)  # Uso inicial
            de CPU
```

```
20          io_inicial = psutil.disk_io_counters()  # Operaciones de
                entrada/salida iniciales
21          hilos_iniciales = proceso.num_threads()  # Numero inicial de
                hilos
22
23          # Cargar los datos desde el archivo CSV
24          data = pd.read_csv(csv_file)
25          print("Datos cargados correctamente.")
26
27          # Verificar si el archivo contiene la columna objetivo
28          if Enfermo not in data.columns:
29              raise ValueError(f"La columna objetivo '{Enfermo}' no esta
                    en los datos.")
30
31          # Separar caracter sticas (X) y etiquetas (y)
32          X = data.drop(columns=[Enfermo])
33          y = data[Enfermo]
34
35          # Asegurar que no haya valores nulos
36          if X.isnull().sum().any() or y.isnull().sum().any():
37              raise ValueError("Los datos contienen valores nulos. Por
                    favor, l mpialos antes de continuar.")
38
39          best_model = None
40          best_accuracy = 0
41          best_random_state = None
42          accuracy_list = []  # Lista para almacenar las exactitudes
43
44          # Probar diferentes valores de random_state en el rango dado
45          for random_state in random_state_range:
46              print(f"\nProbando con random_state={random_state}")
47
48              # Dividir los datos en conjuntos de entrenamiento y prueba
49              X_train, X_test, y_train, y_test = train_test_split(X, y,
                    test_size=0.25, random_state=random_state)
50
51              # Crear el modelo SVM
52              svm_model = SVC(kernel='poly', C=100, degree=2,
                    random_state=random_state)
53
54              # Entrenar el modelo
55              svm_model.fit(X_train, y_train)
56
57              # Realizar predicciones
58              y_pred = svm_model.predict(X_test)
59
60              # Evaluar el modelo
```

```
61              accuracy = accuracy_score(y_test, y_pred)
62              accuracy_list.append(accuracy)  # Guardar la exactitud en
                    la lista
63              print(f"Exactitud: {accuracy:.2f}")
64
65              # Actualizar el mejor modelo si es necesario
66              if accuracy > best_accuracy:
67                  best_accuracy = accuracy
68                  best_model = svm_model
69                  best_random_state = random_state
70
71          # Mostrar resultados del mejor modelo
72          print("\n========== Resultados del mejor modelo ==========")
73          print(f"Mejor random_state: {best_random_state}")
74          print(f"Mayor exactitud: {best_accuracy:.2f}")
75
76          # Evaluar el mejor modelo en detalle
77          X_train, X_test, y_train, y_test = train_test_split(X, y,
                test_size=0.25, random_state=best_random_state)
78          y_pred = best_model.predict(X_test)
79          print("Reporte de clasificacion del mejor modelo:")
80          print(classification_report(y_test, y_pred))
81
82          # Calcular el promedio de las exactitudes
83          avg_accuracy = np.mean(accuracy_list)
84          print(f"\nPromedio de exactitudes para todos los random_state:
                {avg_accuracy:.2f}")
85
86          # Detener medicion de recursos
87          fin_tiempo = time.time()
88          memoria_final = proceso.memory_info().rss / 1024 / 1024  #
                Memoria final en MB
89          uso_memoria = tracemalloc.get_traced_memory()
90          cpu_final = psutil.cpu_percent(interval=None)
91          io_final = psutil.disk_io_counters()
92          hilos_finales = proceso.num_threads()
93          temperatura = psutil.sensors_temperatures() if hasattr(psutil,
                "sensors_temperatures") else None
94          tracemalloc.stop()
95
96          # Calcular diferencias en recursos
97          tiempo_total = fin_tiempo - inicio_tiempo
98          memoria_usada = (uso_memoria[0] / 1024 / 1024)  # Memoria
                actual usada en MB
99          memoria_maxima = (uso_memoria[1] / 1024 / 1024)  # Memoria
                maxima en MB
```

```
100         lecturas_dispositivo = io_final.read_count -
               io_inicial.read_count
101         escrituras_dispositivo = io_final.write_count -
               io_inicial.write_count
102
103         # Mostrar resumen de recursos utilizados
104         print("\n--- Recursos utilizados ---")
105         print(f"Tiempo total de ejecucion: {tiempo_total:.2f} segundos")
106         print(f"Memoria usada: {memoria_usada:.2f} MB")
107         print(f"Memoria maxima: {memoria_maxima:.2f} MB")
108         print(f"Memoria inicial del proceso: {memoria_inicial:.2f} MB")
109         print(f"Memoria final del proceso: {memoria_final:.2f} MB")
110         print(f"Uso inicial de CPU: {cpu_inicial:.2f}%")
111         print(f"Uso final de CPU: {cpu_final:.2f}%")
112         print(f"Hilos iniciales: {hilos_iniciales}")
113         print(f"Hilos finales: {hilos_finales}")
114         print(f"Lecturas en disco: {lecturas_dispositivo}")
115         print(f"Escrituras en disco: {escrituras_dispositivo}")
116         if temperatura:
117             for name, entries in temperatura.items():
118                 for entry in entries:
119                     print(f"Temperatura {name} - {entry.label or
                           'desconocido'}: {entry.current} C ")
120
121         return best_model  # Devolver el mejor modelo entrenado
122     except FileNotFoundError:
123         print(f"El archivo '{csv_file}' no fue encontrado.")
124     except ValueError as e:
125         print(f"Error en los datos: {e}")
126     except Exception as e:
127         print(f"Ocurrio un error: {e}")
128
129 # Ejemplo de uso
130 csv_file = "Ubicacion del archivo"
131 target_column = "Enfermo"
132 random_state_range = range(0, 100)  # Valores de random_state de 0 a 100
133 modelo_svm_optimo = find_best_svm_model(csv_file, target_column,
       random_state_range)
```