



UNIVERSIDAD DEL SINÚ
Elías Bechara Zainúm
Seccional Cartagena

COMPARACIÓN DE MODELOS PREDICTIVOS DE RIESGO DE
HIPERTENSIÓN EN LA POBLACIÓN DE CARTAGENA USANDO APRENDIZAJE
AUTOMÁTICO

Presentado por:

JOHN JAIRO MELÉNDEZ CARABALLO
RODOLFO ALQUERQUE SUAREZ

UNIVERSIDAD DEL SINÚ ELÍAS BECHARA ZAINÚM SECCIONAL CARTAGENA
ESCUELA DE INGENIERÍA DE SISTEMAS

Octubre 2018



COMPARACIÓN DE MODELOS PREDICTIVOS DE RIESGO DE
HIPERTENSIÓN EN LA POBLACIÓN DE CARTAGENA USANDO APRENDIZAJE
AUTOMÁTICO

Trabajo de grado presentado como requisito para optar el título de
INGENIERO DE SISTEMAS

Asesor disciplinar

EUGENIA ARRIETA RODRÍGUEZ

Asesor metodológico

EUGENIA ARRIETA RODRÍGUEZ

UNIVERSIDAD DEL SINÚ ELÍAS BECHARA ZAINÚM SECCIONAL CARTAGENA
ESCUELA DE INGENIERÍA DE SISTEMAS
CARTAGENA-COLOMBIA
Octubre 2018

Acta de Calificación y aprobación

Notas de aceptación

Director de escuela

Director de investigaciones

Firma de jurado

Firma de jurado

Cartagena de indias, 2018

**EL DIRECTOR DE INVESTIGACIONES DE LA UNIVERSIDAD DEL SINU “ELIAS
BECHARA ZAINUM” SECCIONAL CARTAGENA**

HACE CONSTAR QUE:

En Cartagena, a los 6 días del mes de noviembre del 2018, en la Oficina de la Dirección de Investigaciones de la Universidad, se aprobó por el jurado y se realizó la sustentación del Trabajo de Grado titulado “COMPARACIÓN DE MODELOS PREDICTIVOS DE RIESGO DE HIPERTENSIÓN EN LA POBLACIÓN DE CARTAGENA USANDO APRENDIZAJE AUTOMÁTICO” que se desarrolló bajo la dirección del Ingeniero EUGENIA ARRIETA RODRÍGUEZ presentado por los egresados JOHN JAIRO MELENDEZ CARABALLO, RODOLFO ARQUERQUE SUAREZ.

Los jurados designados fueron los ingenieros ANDY CABARCAS SIERRA y AISNER MARRUGO JULIAO.

Teniendo en cuenta la aprobación emitida, se encuentra que los estudiantes han cumplido con los requisitos de presentación y sustentación del trabajo de investigación, exigidos por el programa de INGENIERÍA DE SISTEMAS, Resolución 0178 de 15 de marzo de 2010.

Se expide esta constancia a los 15 días del mes noviembre de del 2018.

DIRECCIÓN DE INVESTIGACIONES
Universidad del Sinú

COORDINADOR DE INVESTIGACIONES
Escuela de Ingeniería de Sistemas

Cartagena de Indias, 14 de noviembre de 2018

Director

MARÍA CLAUDIA BONFANTE RODRÍGUEZ

Directora de la Escuela de Ingeniería de Sistemas

Universidad del Sinú

Cordial saludo.

La presente comunicación con el fin de manifestar mi conocimiento y aprobación del trabajo de grado titulado **COMPARACIÓN DE MODELOS PREDICTIVOS DE RIESGO DE HIPERTENSIÓN EN LA POBLACIÓN DE CARTAGENA USANDO APRENDIZAJE AUTOMÁTICO**, elaborada por los estudiantes JOHN JAIRO MELENDEZ CARABALLO con número de cédula de ciudadanía 1.143.359.815 y RODOLFO ARQUERQUE SUAREZ con número de cédula de ciudadanía 1.143.353.922, presentado como requisito para optar al título de Ingeniería de Sistemas.

Cordialmente,

EUGENIA LUZ ARRIETA RODRÍGUEZ

Coordinadora de investigación de ingeniería de sistemas.

DEDICATORIAS

En primera instancia este nuevo logro se lo dedico a Dios por hacerme la persona que tanto quise ser, él es perfecto y misericordioso gracias por todo.

A mi madre, padre, hermanos, primos y demás familiares que siempre creyeron en mí, todos son autores de este triunfo, por nunca abandonarme y ayudarme a alcanzar esta meta. Este triunfo es de ustedes y para ustedes.

De todas las personas que estoy agradecido hay una en la que nunca tendré lo suficiente para agradecerle, esa es mi madre Justina Caraballo Escobar, ella es quien más goza de estos momentos de la vida por eso es mi luz para seguir adelante, gracias por existir.

Le agradezco a todas las personas que una vez fueron mis tutores de mi carrera y que siempre aportaron sus conocimientos y experiencias para que cada día aprendiera a ser mejor en todo.

Para mis amigos, colegas y compañeros le dedico esta meta que me da gran alegría y emoción saber que valió la pena pasar por tanto y dedicarse para cumplirlo.

Este triunfo se lo dedico a mi madre Carmen Cecilia Suarez Rodríguez y a mi padre Oswaldo Rafael Alquerque Borja quien desde el cielo siempre cuida y me anima a seguir adelante en esta hermosa profesión las personas que siempre estuvieron ahí para acompañarme en todos los momentos buenos malos

Gracias.

AGRADECIMIENTOS

Muy agradecidos con nuestros tutores Eugenia Arrieta, John Arrieta, Rafael López, Jonathan Berthel, Helmut Maldonado, Claudio Aldana, Rafael Monterrosa y a todo el cuerpo Docente quienes nos extendieron su mano para brindarnos apoyo y quienes forjaron nuestro camino con mucha sabiduría y dándonos fortaleza cuando sentíamos que todo estaba perdido, gracias a ustedes hemos logrado esta meta muy importante en la vida de cada uno de nosotros y de nuestras familias, sus consejos nos convirtieron en mejores personas, en todos unos profesionales y al igual que ustedes aplicaremos en nuestras vida esa enseñanza que nos quedó...

Muchas gracias por todo y que Dios derrame muchas bendiciones sobre ustedes y sus familias

TABLA DE CONTENIDO

INTRODUCCIÓN	17
1 DISEÑO METODOLÓGICO	20
1.1 PLANTEAMIENTO DEL PROBLEMA	20
1.1.1 Descripción del problema	20
1.1.2 Justificación	22
1.1.3 Formulación del problema	23
1.1.4 Objetivos	23
1.2 ESTADO DEL ARTE	24
1.3 MARCOS DE REFERENCIA	26
1.3.1 Marco teórico	26
1.3.2 Marco conceptual	38
1.4 METODOLOGÍA	42
1.4.1 Línea de investigación	43
1.4.2 Tipo de investigación	43
1.4.3 Metodología descriptiva	43
2 ANÁLISIS DEL PROBLEMA	45
2.1 REQUISITOS ESPECÍFICOS IEEE-830	45
2.2 REQUISITOS FUNCIONALES	46
2.2.1 Algoritmos	46
2.3 REQUISITOS NO FUNCIONALES	46
2.3.1 ENTORNO DESARROLLO	46
3 DISEÑO DE LA SOLUCIÓN	48
3.1 Organización de la información	48
3.1.1 Recolectar la información de los datos que se necesitan	48
3.1.2 Preparación del set de datos.	48
3.2 Preparación del entorno de desarrollo.	48
3.2.1 Instalación de Anaconda.	48
3.2.2 Instalación de librerías.	49
3.3 Implementación de técnicas de IA.	49

3.3.1	Análisis de las técnicas.	49
3.3.2	Desarrollo de algoritmo de selección de variables	49
3.3.3	Normalización del set de datos	50
3.4	Desarrollo de algoritmos	50
3.4.1	Desarrollo de algoritmo de regresión logística.	50
3.4.2	Desarrollo de algoritmo de máquinas de soporte vectorial	50
3.5	Comparación de los modelos	50
3.5.1	Análisis de resultados	50
3.5.2	Tabla comparativa de los resultados obtenidos	51
4	DESARROLLO	52
4.1	Colecta y preparación del set de datos.	52
4.1.1	Colecta de datos	52
4.1.2	Preparación del formato	53
4.1.3	Técnica de selección de variable	53
4.2	Preparación del entorno	54
4.2.1	Instalación de Anaconda	54
4.3	Implementación de algoritmos	56
4.3.1	Importaciones necesarias	56
4.3.2	Métodos de manejo de data set	57
4.3.3	Técnicas de Machine Learning	59
4.3.4	Selección de variables.	62
4.3.5	División del set de datos.	62
4.3.6	Modelo de regresión logística.	63
4.3.7	Modelo de Máquinas de Soporte Vectorial.	66
5.	COMPARACIÓN DE MODELOS	70
5.1	RESULTADOS DE LAS VALIDACIONES DE LOS MODELOS	70
5.1.1	Características de Máquinas de soporte vectorial y Regresión logística	70
6	RECOMENDACIONES	83
7	CONCLUSIONES	84
	Bibliografía	85
	ANEXOS	87

LISTAS DE TABLAS

	Pág.
Tabla 1 Matriz de confusión	42
Tabla 2 Requerimiento - RF01 Set de datos	48
Tabla 3 Requerimiento -RNF01 Entorno	49
Tabla 4 Resultados de técnicas de selección de variables	72
Tabla 5 Precisión de los subconjunto StandardScalers	72
Tabla 6 Precisión de los subconjunto Normalizer	73
Tabla 7 Precisión de los subconjunto MaxMinScaler	73
Tabla 8 Precisión de los subconjuntos	73
Tabla 9 Resultados de precisión de todos los subconjuntos en modelo SVC	77
Tabla 10 Resultados de precisión en todos los subconjuntos normalizados	78
Tabla 11 Resultados de precisión, sensibilidad y especificidad (SVC).	80

LISTA DE FIGURAS

	Pág.
Figura 1.1 Aprendizaje supervisado	28
Figura 1.2 Aprendizaje automático.	28
Figura 1.3 Gráfica de selección de variables	31
Figura 1.4 Matriz de selección de variables	32
Figura 1.5 Función Sigmoide	33
Figura 1.6 Hiperplano separación óptimo	36
Figura 1.7 Márgenes del hiperplano	37
Figura 1.8 Sobreajuste Overfitting	40
Figura 1.9 Equilibrio del Aprendizaje	41
Figura 1.10 Gráfica de ROC	43
Figura 4.1 Set de datos	53
Figura 4.2 Formato set de datos	54
Figura 4.3 instalador Anaconda	56
Figura 4.4 Versiones del entorno	56
Figura 4.5 Importaciones y Librerías Python	57
Figura 4.6 Métodos de manejo de datos	57
Figura 4.7 Encabezados y características de los datos	58
Figura 4.8 Inicialización de variables globales	58
Figura 4.9 Normalización de los datos	59
Figura 4.10 Conversión de tipo de datos	60
Figura 4.11 Gráfica valores de características	61
Figura 4.12 Gráficas comportamiento características.	62
Figura 4.13 División del set de datos	63
Figura 4.14 Modelo de regresión logística	64
Figura 4.15 Validación del modelo regresión logística	64
Figura 4.16 Matriz de confusión	65
Figura 4.17 Medición del modelo regresión logística	65
Figura 4.18 curva ROC (modelo regresión logística)	65
Figura 4.19 Gráfica curva ROC (modelo regresión logística)	66
Figura 4.20 Modelo Máquinas de Soporte Vectorial	66
Figura 4.21 Evaluación del modelo Máquinas de Soporte Vectorial	67
Figura 4.22 Predicción del modelo SVC	67
Figura 4.23 Matriz de Confusión Máquinas de Soporte Vectorial	68
Figura 4.24 Medición del modelo Máquinas de soporte vectorial	68
Figura 4.25 Gráfica curva de ROC	69
Figura 5.1 Selección de variables ExtraTreesClassifier	71
Figura 5.2 Selección de variables LinearSVC	71
Figura 5.3 Selección de variables Pipeline	71
Figura 5.4 Comparación de modelos en subconjuntos	74
Figura 5.5 Comparación de modelos en subconjuntos MaxMInScaler	75

Figura 5.6 Comparación de modelos en subconjuntos normalizados con StandardScaler	76
Figura 5.7 Comparación de modelos en subconjuntos normalizados con Normalizer	77
Figura 5.8 Comparación de precisiones normalizadas con cada subconjunto	78
Figura 5.9 Comparación de precisión del modelo regresión logística en todos los subconjuntos	79
Figura 5.10 Comparación de resultados de medición con todos los subconjuntos en SVC	80
Figura 5.11 resultados de mediciones del modelo regresión logística en todos los subconjuntos	80
Figura 5.12 Comparación de mediciones del modelo regresión logística en todos los subconjuntos	81

LISTA DE ECUACIONES

	Pág.
Ecuación 1.1 Desigualdad de Hoeffding	29
Ecuación 1.2 Desigualdad de Hoeffding	30
Ecuación 1.3	32
Ecuación 1.4	33
Ecuación 1.5	33
Ecuación 1.6	34
Ecuación 1.7	34
Ecuación 1.8	34
Ecuación 1.9	34
Ecuación 1.10	34
Ecuación 1.11	35
Ecuación 1.12	37
Ecuación 1.13	37
Ecuación 1.14	37
Ecuación 1.15	38
Ecuación 1.16	38
Ecuación 1.17	38
Ecuación 1.18	38
Ecuación 1.19	39
Ecuación 1.20	43
Ecuación 1.21	44
Ecuación 1.22	44

LISTA DE ANEXOS

	Pág.
ANEXO A. Presupuesto del proyecto	87

RESUMEN

Para la elaboración de este trabajo se realizaron en parte investigaciones distintas unas más profundas que las otras y se analizaron varias propuestas para la realización de algoritmos que utilizan inteligencia artificial, entonces es cuando se encuentra que es posible aplicar el machine learning para dar soluciones de ayuda en algunos casos. De todas las ideas se halló un proyecto donde aplicamos aprendizaje automático para casos de hipertensión en la ciudad de Cartagena, para la implementación de las diferentes técnicas, primero se recopiló toda la información necesaria de varios conjuntos de datos de pacientes que se unificaron para su preparación y selección de características. Para la salud estas herramientas son bastante útiles donde podemos aplicar técnicas de machine learning para detectar o clasificar personas que están en riesgos de hipertensión. En esta parte se inició la investigación de objetivos, antecedentes y sobre todo el conjunto de datos para los modelos predictivos, pero para hacer el proyecto más alcanzable se define como meta final realizar una comparación de los modelos de aprendizaje automático desarrollados a través de la librería sklearn en Python para determinar cuál es el modelo más óptimo para obtener mejores resultados en términos de predicción.

Todos los algoritmos, técnicas o métodos fueron desarrollados bajo el lenguaje de programación Python, para el cual se utilizó la plataforma especialmente utilizada para laboratorios de ciencia de datos e inteligencia artificial, esta se llama anaconda la cual es muy fácil de manejar y ofrece muchas ayudas las cuales hacen más fácil la codificación.

Anaconda incluye toda la suite de paquetes básicos que se van a requerir para el análisis y creación de los modelos predictivos, no es necesario instalar librerías externas ya que de todas formas la plataforma maneja las dependencias de manera organizada. Primero se recolectó toda la información con la que se va a trabajar en los algoritmos, luego se debe organizar para presentarla de manera unificada en un archivo que se va a leer a través de métodos y librerías.

Se realizaron todos los algoritmos y técnicas para la preparación de los datos y el manejo entre los diferentes modelos, del conjunto de datos original se realizó una segmentación de datos dividiendo los datos en entrenamiento y prueba que sirven para el entrenamiento y validación de los modelos, se realizaron todas las pruebas posibles para evaluar los modelos lo cual se hace para identificar los comportamientos y efectos. Luego se evaluaron los subconjuntos de datos obtenidos para comparar cada uno de los modelos seleccionados, realizando un análisis de clasificadores de acuerdo a las variables seleccionadas para elegir el mejor modelo, por último, concluyo cuál es el modelo más óptimo de los comparados. El más apto para asumir la responsabilidad de predecir casos de hipertensión, sabiendo que los modelos de predicción no son 100% certeros son datos que nos ayudarán en gran parte a tomar decisiones que pueden prevenir riesgo de casos de hipertensión en la salud.

INTRODUCCIÓN

La inteligencia artificial es una rama de estudio científico de la ingeniería de sistemas y otras ciencias que buscan reemplazar el trabajo del hombre en computadoras inteligentes para trabajo y automatización de procesos buscando la mejora continua en los trabajos del ser humano. En inteligencia artificial se encuentra una rama que se dedica a el aprendizaje en máquinas a través de modelos predictivos utilizando aprendizaje automático el cual funciona de manera utilitaria para el apoyo profesional y el desarrollo humano.

En Cartagena, Bolívar, Colombia, se encuentra en un momento en que la necesidad y las problemáticas ambientales en viviendas de bajo recurso, estén expuestas a enfermedades con riesgos de hipertensión. Hace varios años atrás era difícil el acceso a tecnologías y herramientas que puedan presentarse para la prevención de hipertensión utilizando aprendizaje automático para la predicción de diagnósticos basado en un estudio de datos, actualmente es posible realizar estos cálculos de variables a un nivel de complejidad medida por los especialistas y desarrolladores.

Se realizó un estudio previo para implementar las técnicas de aprendizaje automático en un conjunto de datos, la manera en que se recogieron los datos fue por diagnósticos ya realizados por especialistas en tema de hipertensión en Cartagena.

El aprendizaje automatizado en máquinas es utilizado en profesionales como apoyo para el desarrollo humano en la toma de decisiones y para este caso es prevenir futuros casos de riesgo hipertensión. Esta manera de realizar una profesión es totalmente legal y válida para la prevención de casos con riesgo de hipertensión ya que esta busca dentro de diagnósticos las variables especializadas para la toma de decisiones con el criterio de un especialista en el área, validando los resultados y decisiones tomadas por el modelo que se implemente para su realización. Este aporte busca ayudar a decidir cuál de los modelos y técnicas que se van presentar es el más adecuado para implementarse en proyectos que usen aprendizaje automatizado para prevenir riesgos de hipertensión.

el desarrollo del proyecto se busca comparar modelos de predicción de aprendizaje automatizados supervisado. Este se implementó con un conjunto de datos que fueron resultados de diagnósticos de personas que de alguna otra forma pudieron estar en riesgo de hipertensión es decir pacientes presentaron patologías de riesgo de hipertensión y fueron diagnosticados por un médico especialista en el área clínica de la salud humana.

Existe un conflicto en la sociedad de investigadores que aplican modelos de aprendizaje automatizado para la detección y prevención de casos en personas con riesgos de hipertensión y en Cartagena. Actualmente no se han publicados modelos que busquen la prevención de hipertensión con datos de diagnósticos reales de personas. en este trabajo se realizó una implementación de modelos predictivos, los cuales fueron comparados para decidir cuál de los dos modelos es el más apto para desarrollar diagnósticos más confiables y cercanos a la precisión de acuerdo a sus parámetros y aplicación realizados por el especialista del área de salud.

El análisis de los resultados de la comparación de los modelos es totalmente auténtico y dirigido por profesionales que manejan el tema de su área para el criterio adecuado y puntual, también se argumenta de ingenieros de sistemas que manejan el área de desarrollo de software y las técnicas utilizadas en el aprendizaje automatizado. De esta manera se declara que este proyecto y su metodología y resultados son auténticos y bastante importante para la sociedad que busca el cuidado de la humanidad.

Para la construcción del proyecto participaron ingenieros, médicos y estudiantes con diversos roles que contribuyen al resultado del proyecto de manera directa, su aporte es de gran importancia como la todos los autores debido a que su experiencia en elaboración de proyectos e investigaciones, gracias a el contacto con el médico especialista del área de la salud con conocimientos en riesgos de hipertensión con sus set de datos de diagnósticos realizados a pacientes que pueden padecer de un estado de riesgo, así mismo su conocimiento para la elaboración de los dos modelos de aprendizaje automatizado supervisado, apoyando enormemente el objetivo principal y específicos del proyecto.

El equipo de apoyo investigativo es indispensable para la construcción del proyecto su participación es directa con la concepción y el diseño de que se entregará como resultado de cada uno de los objetivos específicos de este proyecto.

Los investigadores especialistas del área de la salud obedecen las medidas de prevención a la información dada para los procesos que incluyen conocimientos médicos que están dentro del proyecto y de esta manera también se protege la integridad y confidencialidad de los resultados del set de datos obtenidos de los diagnósticos de pacientes reales para la construcción de los dos modelos, es decir el objetivo que se busca de cada modelo de aprendizaje automatizado es que los resultados sean precisos y con el más efectivos y no ver cuáles son los pacientes con riesgos de hipertensión, de esta forma se maneja el secreto profesional de los especialistas médicos.

El proyecto comparación de modelos predictivos de riesgo de hipertensión en la población de Cartagena usando Aprendizaje Automático es aporte investigativo para todas las personas que buscan la mejora continua de la humanidad y otros investigadores que les interese, puede ser accedida para consulta de manera autorizada sólo con solicitud aprobada a los actores principales, igualmente los actores que de alguna otra forma tendrán el acceso y permisos para trabajar con los resultados de la investigación, es decir los asesores y médicos involucrados tendrán copia original del desarrollo del proyecto, como agradecimiento a su participación y aporte al proyecto.

El proyecto es dirigido hacia la Universidad del Sinú para presentar como proyecto o tesis de grado por eso no tiene un coste de publicación en ninguna plataforma para sociedad.

Se reservan todos los derechos a los autores principales y autores secundarios importantes en la investigación y a la Universidad del Sinú de poder copiar, modificar y utilizar los aportes y resultados del proyecto.

Los aportes de los estudios de investigaciones realizadas por los ingenieros y médicos son voluntarios, la información que es sugerida por el médico es confidencial para la aplicación de los modelos que se construyeron, en ninguna parte del proyecto se

compromete la información de los pacientes del set de datos requerido por el médico, aunque ya se definió que este no era el objetivo del proyecto.

En los campos de la ciencia informática y la salud se han estado realizando diferentes estudios profesionales y científicos para adelantar a posibles trastornos como es el riesgo de hipertensión, aplicando diferentes tecnologías e inteligencia artificial en los datos de pacientes que pueden estar en posibles condiciones de padecer riesgos [1].

Existe la necesidad en Cartagena de que los médicos cuente con herramientas y tecnologías de apoyo para la toma de decisiones de los diagnósticos de pacientes que tengan riesgos de trastornos hipertensivos y que los otros procesos de recopilación de información se evidencian en registros para investigaciones en prevención. El aprendizaje automatizado en máquinas es el apoyo y herramientas tecnológica que implementa inteligencia artificial para ayudar médicos a desarrollar con más eficacia en la profesión en el área de la salud y que con la comparación de usos de modelos algoritmos capaces de aprender, seleccionar y clasificar los datos en computadoras y aportar a investigadores, desarrolladores, médicos y emprendedores un gran avance de opiniones y aportes para proyectos próximos relacionados con aprendizaje automatizado supervisado.

Actualmente existen muchas herramientas y tecnologías que sirven para dar aportes a las ciencias de la salud y la medicina, además que ya ha habido personas y grupos que debido a eso cada vez es más creciente la comunidad que tiene la necesidad del estudio de la información a través de la tecnología.

Por lo anterior se propone hacer una comparación del uso e implementación de algoritmos de machine learning aplicado con el lenguaje de programación Python y una base de datos donde se encuentran la información de varias personas con sus datos.

Nuestro país está en mora de aplicar de manera más amplia la inteligencia artificial en el campo de la medicina. Con los avances en computación de hoy y la ubicuidad de equipos de cómputo relativamente potentes, a través del despliegue de algoritmos de Machine Learning para la solución de este tipo de problemas de diagnóstico y muchos otros problemas relacionados a la práctica médica, se puede impactar de manera realmente sustancial la calidad de los servicios médicos, a muy bajo costo [1].

1 DISEÑO METODOLÓGICO

1.1 PLANTEAMIENTO DEL PROBLEMA

1.1.1 Descripción del problema

A pesar de lograr avances en la salud, las complicaciones relacionadas con la prevención de pacientes con riesgos de hipertensión siguen siendo un importante problema de salud pública en el mundo. En la actualidad existen muchas tecnologías capaces de ayudar a prevenir los trastornos de hipertensión en las personas, aplicando técnicas de predicción utilizando machine learning para analizar las patologías.

Varios estudios ya adelantados dicen que la hipertensión arterial provoca cada año 7,5 millones de muertes, el 13% del total de defunciones que se producen a nivel global, según la World Health Organization (WHO), en Latinoamérica, el 13% de las muertes y el 5,1% de los años de vida ajustados por discapacidad (AVAD) pueden ser atribuidos a la hipertensión. La prevalencia ajustada para la edad de la hipertensión en la población adulta general en diferentes países de Latinoamérica (encuestas nacionales o muestreos sistemáticos aleatorizados) varía entre el 26 al 42% [2].

La evidencia epidemiológica sugiere que la hipertensión es un factor de riesgo para el desarrollo de diabetes. En las poblaciones diabéticas, la prevalencia de la hipertensión es 1,5 a 3 veces mayor que en no diabéticos de la misma franja etaria. En la diabetes tipo 2, la hipertensión puede ya estar presente en el momento del diagnóstico o inclusive puede preceder a la hiperglucemia evidente (4,5). Son muchas las complicaciones que se pueden presentar si no existe un control de la hipertensión arterial como enfermedades cardio-cerebrovasculares, insuficiencia renal, deterioro cognitivo por microaneurismas, entre otros (6–11).

En Colombia, según la Organización Panamericana de la Salud (OPS), la prevalencia de HTA en la población mayor de 15 años es del 12,6%, constituyéndose en el principal factor de riesgo para enfermedades cardiovasculares (12), a esto se le suma la falta de un sistema de información confiable que permita conocer en tiempo real la problemática de salud, seguimiento, control y adherencia al tratamiento de los pacientes con Hipertensión Arterial.

En Cartagena, son pocas las organizaciones de salud que garantizan la correcta colección y control de la información que el gobierno usa para generar perfiles epidemiológicos, estudios clínicos y reportes que ayudan a crear políticas para la administración de la salud de la población y el desarrollo social. Por otra parte, se ha incrementado en un 35% en los últimos 10 años la población en Cartagena de Indias, y desafortunadamente este aumento ha ocurrido en áreas subdesarrolladas, áreas sin instalaciones de salud y pobre acceso a tratamientos de salud específicos. Este crecimiento de la población afecta a la planeación de la salud en Cartagena y las instituciones de salud de la ciudad. Por ley, el gobierno colombiano debe ser capaz de garantizar y proveer servicios de salud primarios, planes de prevención de enfermedades, programas para la salud de la conducta, y mejoras en la calidad de vida de la población [3] [4].

En adición a lo anteriormente citado y debido la baja capacidad de atención de algunos centros de salud en Cartagena para tratar y retener a los pacientes, los profesionales de la salud están permitiéndoles ser trasladados en sus hogares para su recuperación o para seguir con su tratamiento. Y la necesidad de monitorear el estatus y condiciones de salud de estos pacientes es inadecuada con la presencia de complicaciones que pudieron haberse prevenido con una supervisión ajustada a las necesidades y condiciones del paciente [3] [4].

En Colombia en el año 2007 se convocó a un grupo de reconocidos expertos del país para que revisaran la evidencia disponible acerca de diferentes aspectos relacionados con la hipertensión y redactaran un documento (no una revisión del tema) en forma de guía, que permitiera tener claridad sobre esta información y la mejor manera de emplearla en la práctica diaria. El reto de este grupo de expertos estuvo en lograr que las guías se actualicen con frecuencia, a medida que se obtenga información nueva que implique un cambio en la recomendación [3].

Es evidente que la hipertensión es una de las enfermedades que se encuentra dentro de las diez primeras causas de mortalidad, es un problema que se presenta tanto en población joven como en adultos y ancianos. Este tipo de enfermedades pueden conducir a complicaciones mayores, como es el caso de la preclamsia que se presenta en el embarazo y pone en riesgo de muerte tanto a la madre como al bebé.

Debido a la problemática de falta de modelos de predicción se suma que además en Colombia hay poca disponibilidad de datos epidemiológicos sobre la preeclampsia [6], se suma a esto que en la población se aplican modelos propuestos para otro tipo de población [5], lo cual no quiere decir que no se hayan hecho grandes avances en los últimos años en lo relacionado con la comprensión como en el control de la enfermedad desde el punto de vista etiológico, fisiopatológico, clínico, epidemiológico y por ende en el campo de la salud pública por medio de la implementación de guías de práctica clínica, campañas de educación y actualización sobre el diagnóstico y tratamiento oportuno [6]. La hipertensión arterial se considera una enfermedad compleja multicausal ya que implica en sus procesos de presentación un condicionamiento epidemiológico y ambiental los cuales interaccionan entre sí, a los cuales se le suma factores intrínsecos como los genéticos o inmunológicos, incluyendo el acceso a los servicios de salud [7].

Se considera que los modelos actuales de tamizaje o predicción, la tasa de detección de trastornos hipertensivos con estos factores intrínsecos y extrínsecos como única herramienta de predicción es muy baja, cercana al 30% [8] [7].

Las investigaciones actuales se limitan a describir los factores clínicos y epidemiológicos en el manejo de la enfermedad [8], son pocos los estudios que muestran una correlación entre la prevalencia de los factores considerados de riesgos con las características clínicas, manejo y tratamiento de las pacientes [10].

Por consiguiente, las organizaciones de la salud podrían beneficiarse de utilizar estos dispositivos de control y monitoreo en sus pacientes para poder monitorearlos en tiempo real. Todos estos datos generan reportes y la información será almacenada y tratada en plataformas seguras en la nube. Los avances en la colección de datos a través de sensores y su procesamiento permiten la realización de analíticas de datos en tiempo real, liderando grandes avances en el monitoreo y control de pacientes, y, la forma como la salud pública utiliza esta información para generar programas proactivos de tratamiento y atención a la población.

1.1.2 Justificación

Es posible que existan modelos de predicción que utilicen el aprendizaje automatizado en computadores y tecnologías, pero no es posible tener resultados iguales en diferentes modelos. A partir de la necesidad de escoger modelos que mejor califiquen y que por alguna razón sean más precisos e inteligentes se elabora una comparación de dos

modelos que buscan obtener la mejor precisión para la detección de riesgos de hipertensión en la ciudad de Cartagena.

Decidir y elegir el uso de un modelo para un uso específico ya sea de una aplicación o estudio es mucho más conveniente desde una comparación de modelos de aprendizaje automatizado similares.

Teniendo en cuenta la enorme y creciente necesidad de la medicina para tomar decisiones correctas y oportunas de los pacientes, se propone desde el campo de la ingeniería algunas soluciones tecnológicas que generen apoyo al profesional durante el proceso de atención en salud. Se hace necesaria la construcción de una herramienta que permita la predicción de enfermedades Crónicas no transmisibles junto con la opción de generar alertas cuando algunos signos o grupo de signos se encuentren fuera del rango normal.

Una de las ventajas de la presente investigación está en propender por el diseño de un modelo predictivo que contribuya al diseño de guías de diagnóstico, tratamiento y prevención, además como el uso racional de los recursos cada vez más escasos de salud y faciliten la implementación de procesos administrativos, científicos y logísticos de las patologías o condiciones de salud abordadas.

1.1.3 Formulación del problema

¿Cuál de los dos modelos supervisados de machine learning se ajusta más favorablemente en el desarrollo de un modelo predictivo de casos de hipertensión en la población cartagenera?

1.1.4 Objetivos

Objetivo general

Analizar dos modelos de predicción para riesgo de hipertensión en pacientes de ciudad de Cartagena usando técnicas de machine learning.

Objetivos específicos

- Identificar las variables asociadas a trastornos hipertensivos teniendo en cuenta las recomendaciones de estudios anteriores, para seleccionar solo las variables que históricamente han incidido en estas patologías.

- Construir un conjunto de datos que permita el entrenamiento de los clasificadores o modelos predictivos de riesgos de hipertensión en la ciudad de Cartagena.
- Implementar los algoritmos de aprendizaje automático supervisado que permitan obtener un buen resultado en términos de sensibilidad y precisión mediante el entrenamiento y validación de los algoritmos.
- Comparar resultados de las técnicas aplicadas en términos de sensibilidad, precisión y especificidad para determinar cuál de las técnicas es más efectiva para la predicción de trastornos hipertensivos en la población de Cartagena.

1.2 ESTADO DEL ARTE

Aplicaciones de las Técnicas de Machine Learning en la Medicina, varias investigaciones demuestran la importancia que han tomado estas técnicas para el diagnóstico tratamiento y pronóstico médico, sobre todo cuando se tienen grandes volúmenes de datos. Por los resultados que se han obtenido en estas investigaciones, la comunidad médica a aumentado su confianza en este tipo de herramientas, y en la literatura se encuentran estudios para detección de diferentes tipos de enfermedades, en la especialidad de alergología, en enfermedades de vías respiratorias como la Neumonía, Asma, Bronquiolitis, para enfermedades cardiovasculares, para el descubrimiento de nuevas drogas, para clasificación del dolor, para evaluar tiempos de esperas y asignación de recursos médicos, entre otros, son tantas las aplicaciones en donde ha tomado fuerza las técnicas de machine Learning que hay comunidades dedicadas a alimentar repositorios para contribuir al aprendizaje y evaluación de dichas técnicas como por ejemplo existe un repositorio en línea con bases de datos para utilizar en el aprendizaje automático, mantenido por la Universidad de California en Irvine, que incluye resultados de biopsias de lesiones mamarias, predictores de cardiopatía, registros de supervivencia posquirúrgica y otras 270 bases de datos de diversas disciplinas para aprender [14].

El estudio titulado “Predicción de la evolución hacia la hipertensión arterial en la adultez desde la adolescencia utilizando técnicas de aprendizaje automatizado” de la Universidad Central “Marta Abreu” de las villas facultad de matemática, física y computación.

que tiene como objetivo de analizar dos técnicas: clasificación y de agrupación dando como resultado dos modelos matemáticos enfocado en pacientes que sufren de hipertensión arterial contribuye a la creación de conocimiento a través del análisis de los diagnósticos médicos. (Técnicas de minería de datos aplicadas al diagnóstico de

entidades clínicas, 2012 págs. 174-183) Este antecedente también contribuye en brindar su algoritmo usado para el análisis de datos y poder así delimitar el algoritmo que se realizará y la metodología que se va a usar.

En el trabajo de investigación de (CANDELA Cáceres, 2015) titulado “Proceso de Descubrimiento de Conocimiento para Predecir el Abandono de Tratamiento en una Entidad de Salud Pública” realizado en la ciudad de Lima, cuyo objetivo fue el de automatizar un proceso de descubrimiento de conocimiento para una institución de salud pública que permita determinar el comportamiento de los pacientes con respecto a la continuidad en sus tratamientos. Realizó pruebas con cuatro algoritmos dando como resultado: “Al algoritmo SVM un porcentaje de acierto de 96.4%, siendo el de mayor precisión, al algoritmo de modelos combinados un 95.9%, al algoritmo de árbol de decisión un 83.5%, y al algoritmo de redes neuronales un 53.9%.” Concluyendo que, gracias al algoritmo SVM, se pudo determinar los factores más influyentes como son la edad, la autoestima, los medicamentos suministrados, entre otros y, gracias al algoritmo de árbol de decisión, las reglas asociadas a las categorías de tiempo de duración de la hospitalización. Esta investigación contribuye al uso de minería de datos, el beneficio de emplearlo y permite conocer el funcionamiento de los algoritmos para a partir de ello realizar el modelo propuesto [14].

en el trabajo de grado titulado “Modelo de Machine Learning para la Clasificación de pacientes en términos del nivel asistencial requerido en una urgencia pediátrica con Área de Cuidados Mínimos” que tiene como objetivo de analizar dos técnicas

Identificar los síntomas y los signos que pueden ser usados como predictores de gravedad para clasificar una urgencia pediátrica de acuerdo con el nivel asistencial requerido utilizando modelos de machine learning basados en aprendizaje automático que permitan realizar diagnósticos de pacientes como herramienta de apoyo para los procesos médicos.

Un error médico se puede dar por exceso de servicio cuando el tratamiento no tiene probabilidades que el paciente mejore; por insuficiencia de servicios cuando la atención no es en el tiempo adecuado; y por diagnóstico inapropiado ya que el doctor intuye el posible tratamiento. El diagnóstico es la actividad inicial del médico es el punto de partida siendo una de las actividades principales, se recogen los síntomas para elegir el tratamiento adecuado, además tiene como elementos de entrada: las historias clínicas, el

examen físico, exámenes de laboratorio. En función a ello el doctor infiere el diagnóstico más adecuado. Un error en el diagnóstico es directamente responsable el personal médico ya sea cognitivos debido a una mala recolección de datos o desinformación, también por error del sistema ignorancia o mala práctica de la lex artis. (Errores médicos, 2009)

en el siguiente trabajo de grado titulado “Modelo de minería de datos usando machine learning con reconocimiento de patrones de síntomas y enfermedades respiratorias en las historias clínicas para mejorar el diagnóstico de pacientes en la ciudad de Trujillo 2016.” cuyo objetivo final diagnosticar con mayor precisión utilizando modelos con el modelo de minería de datos y algoritmos de machine learning generando mayor acierto al momento de realizar los diagnósticos médicos pertinentes [14].

en el proyecto colaboración con el Ministerio de Primera Infancia del Gobierno Provincial de Salta Argentina Titulado “Predicción de Embarazo Adolescente con Machine Learning” cuya finalidad específica es utilizar modelos de ML para identificar aquellas adolescentes con mayor riesgo de quedar embarazada.

es una de las ciudades más pobladas de Argentina y capital de la provincia con igual nombre con una población que supera los 500.000 habitantes, su Ministerio de Primera Infancia tiene por misión erradicar la pobreza en la provincia. Con este objetivo en mente, el desarrollo de las capacidades de los individuos es algo fundamental [14].

1.3 MARCOS DE REFERENCIA

1.3.1 Marco teórico

Inteligencia artificial

La Inteligencia artificial o IA es una ciencia que surgió después de la segunda guerra mundial fue nombrada así mismo en el año 1956 como ciencia que estudia la inteligencia en las máquinas. El campo de la inteligencia artificial o IA va más allá de predecir, entender, comprender, sino que también se esfuerza en construir entidades inteligentes capaces de decidir.

La IA abarca en la actualidad una gran variedad de subcampos que va desde un propósito general como el aprendizaje y la percepción a otros específicos diagnósticos de enfermedades, solución de problemas matemáticos y otros. Sintetiza y automatiza tareas

intelectuales humanas de manera universal, que de manera artificial no humana es capaz de razonar puede decirse que son sistemas que piensen como humanos, el nuevo excitante esfuerzo de hacer que los computadores piensen, las máquinas con mentes propia en el más amplio sentido literal. La automatización de actividades que vinculamos con procesos de pensamiento humano, actividades como la toma de decisiones, resolución de problemas y aprendizaje. De la IA se puede decir que se imaginan como sistemas que piensan racionalmente el estudio de las facultades mentales mediante el uso de modelos computacionales y el estudio de los cálculos que hacen posible percibir, razonar y actuar. La IA se considera que todos los sistemas actúan como humanos, estas contienen el arte de desarrollar máquinas con capacidad para realizar funciones que cuando son realizadas por personas requieren inteligencia y el estudio de cómo lograr que los computadores realicen tareas que por el momento los humanos hacen mejor. La IA consiste en sistemas que actúan racionalmente, es decir, la inteligencia computacional es el estudio del diseño de agente inteligentes además está relacionada con la conducta inteligente en artefactos [1].

El aprendizaje automático o aprendizaje de máquinas (del inglés, "Machine Learning") es el subcampo de las ciencias de la computación y una rama de la inteligencia artificial cuyo objetivo es desarrollar técnicas que permitan a las computadoras aprender. Es decir, trata de crear programas capaces de generalizar comportamientos a partir de una información suministrada en forma de ejemplos. Es, por lo tanto, un proceso de inducción del conocimiento. En muchas ocasiones el campo de actuación del aprendizaje automático se solapa con el de la estadística computacional, ya que las dos disciplinas se basan en el análisis de datos. Sin embargo, el aprendizaje automático también se centra en el estudio de la complejidad computacional de los problemas. Este puede ser visto como un intento de automatizar algunas partes del método científico mediante métodos matemáticos [15].

Aprendizaje supervisado

Aprendizaje automático supervisado es una técnica para deducir una función a partir de datos de entrenamiento. Los datos de entrenamiento consisten de dos vectores: uno representa los datos de entrada y el otro los resultados deseados.

La salida de la función puede ser un valor numérico (como en los problemas de regresión) o una etiqueta de clase (como en los de clasificación). En la Figura 1.1 se muestra el objetivo del aprendizaje supervisado el cual es crear una función capaz de predecir el

valor correspondiente a cualquier objeto de entrada válida después de haber visto una serie de ejemplos, los datos de entrenamiento. Para ello, tiene que generalizar a partir de los datos presentados a las situaciones no vistas previamente [2].



Figura 1.1 Aprendizaje supervisado

En la Figura 1.2 se presenta una descripción formal del problema de aprendizaje automático, indicando que dado un conjunto de datos de entrenamiento que son los vectores X (variables de entrada) y Y (respuesta), se realiza un proceso de entrenamiento usando un algoritmo de aprendizaje, con esto se logra encontrar la función $h(x)$ haga una predicción y . Donde la función h se llama hipótesis o modelo.

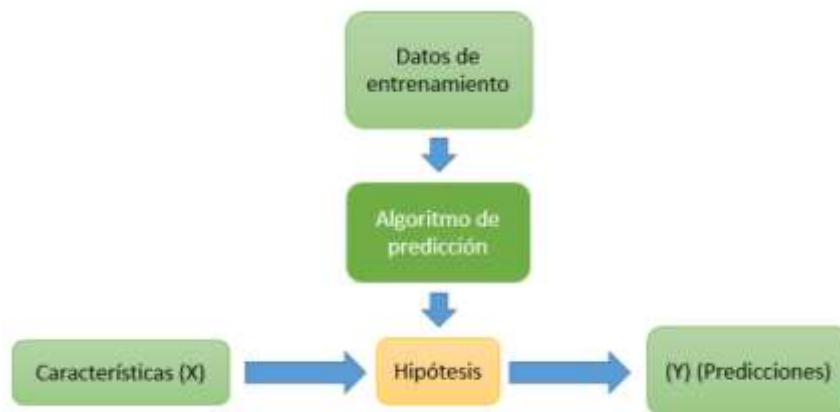


Figura 1.2 Aprendizaje automático.

El algoritmo de aprendizaje debe encontrar los parámetros correctos de la función se realiza un proceso llamado entrenamiento o proceso de aprendizaje. Por ejemplo, si se desea enseñar matemáticas a un grupo de estudiantes, se les entrena en el tema

mediante ejercicios, de esta misma forma se hace con los clasificadores, primero se les enseña mediante casos de ejemplo.

Al igual que en la vida real, a los estudiantes no se les evalúa usando el mismo ejercicio del entrenamiento, si no ejercicios que nunca han visto. De igual forma a los clasificadores tampoco se les evalúa con los mismos casos del proceso de aprendizaje, porque la idea es que pueda predecir correctamente con casos reales. Por ende, el objetivo de un clasificador es que logre predecir casos reales tan bien como lo hizo con los casos de entrenamiento. A esta capacidad de extrapolar correctamente se le llama generalización. Para llegar a lograr la generalización de un clasificador se requiere que en las pruebas el error de predicción sea bajo, a este error se le llama error de pruebas.

Si el error entrenamiento es pequeño, es muy probable que el de pruebas también lo sea, pero esto no es una garantía para el aprendizaje. Para lograr una buena generalización del modelo se recomienda hacer un buen muestreo, tomando una muestra lo suficientemente grande y representativa de la población. Con esto se logra una garantía probabilística de que el aprendizaje es factible. Es decir, que el error de pruebas está cercano al error de entrenamiento, expresado matemáticamente como se muestra en la Ecuación 1.1.

$$P[|u - \mu| > \epsilon] \leq 2e^{-2\epsilon^2 N}$$

Ecuación 1.1 Desigualdad de Hoeffding

Esta fórmula también llamada desigualdad de Hoeffding, donde v corresponde a la fracción de casos mal clasificados en la muestra (en adelante será nombrado error de entrenamiento E_{in}), μ es la fracción de casos que el algoritmo clasifica con error (en adelante será nombrado error de pruebas E_{out}), N es el tamaño de la muestra, es un valor que se desea que sea pequeño. Como el objetivo es que la diferencia entre el E_{in} y el E_{out} sea pequeño, se recomienda aumentar el tamaño de la muestra N . Convirtiendo la desigualdad de Hoeffding en la Ecuación 1.2.

$$P[|E_{in}(h) - E_{out}(h)| > \epsilon] \leq 2e^{-2\epsilon^2 N}$$

Ecuación 1.2 Desigualdad de Hoeffding

En el aprendizaje automático, la función de predicción será nombrada f , y a la hipótesis h . Cuando el algoritmo predice correctamente en un punto o para un caso se dice que $f(x) = h(x)$, cuando la predicción es incorrecta $f(x) \neq h(x)$.

Cuando la variable objetivo, la que se está tratando de predecir, es continua se le llama al problema de aprendizaje un problema de regresión. Cuando se predicción esperada está definida por algunos valores discretos (tales como Si o No), es nombrado un problema de clasificación. Los clasificadores más utilizados son las redes neuronales, las máquinas de soporte vectorial, regresión logística, el algoritmo de los K-vecinos más cercanos, el clasificador bayesiano ingenuo, y los árboles de decisión.

Selección de variables

La selección de características es un proceso que consiste en seleccionar un subconjunto de variables relevantes para la construcción de un clasificador o predictor.

Es frecuente que en un conjunto de datos existen muchos parámetros que tengan relación con la variable de respuesta, haciendo difícil el análisis y la predicción. Para ello se plantea la reducción del número de parámetros y obtener sólo aquellos que presentan mayor variabilidad.

El objetivo principal de la selección de variable es mejorar el rendimiento de la predicción del clasificador y esto a su vez proporciona mayor rapidez del clasificador. La selección de características puede ser utilizada para la reducción de la dimensionalidad de la función, ya sea para mejorar los resultados de la precisión de los estimadores o para aumentar su rendimiento en conjuntos de datos de muy alta dimensión.

Teniendo en cuenta un estimador que asigna ponderaciones a las características (por ejemplo, los coeficientes de un modelo lineal), la función recursiva de eliminación (RFE) es para seleccionar funciones, haciendo más y más pequeños el set de datos recursivamente hasta conseguir el número correcto de variables que mejoran los resultados del clasificador.

En la Figura 1.3 se muestra el resultado de aplicar el algoritmo de selección de variables, donde se encuentra el mayor pico de la gráfica es donde se encuentra el número óptimo de variables que se deben seleccionar, para este caso son tres.

El algoritmo también retorna una matriz con verdaderos y falsos que indica si la variable es seleccionada o no, como se muestra en la Figura 1.4

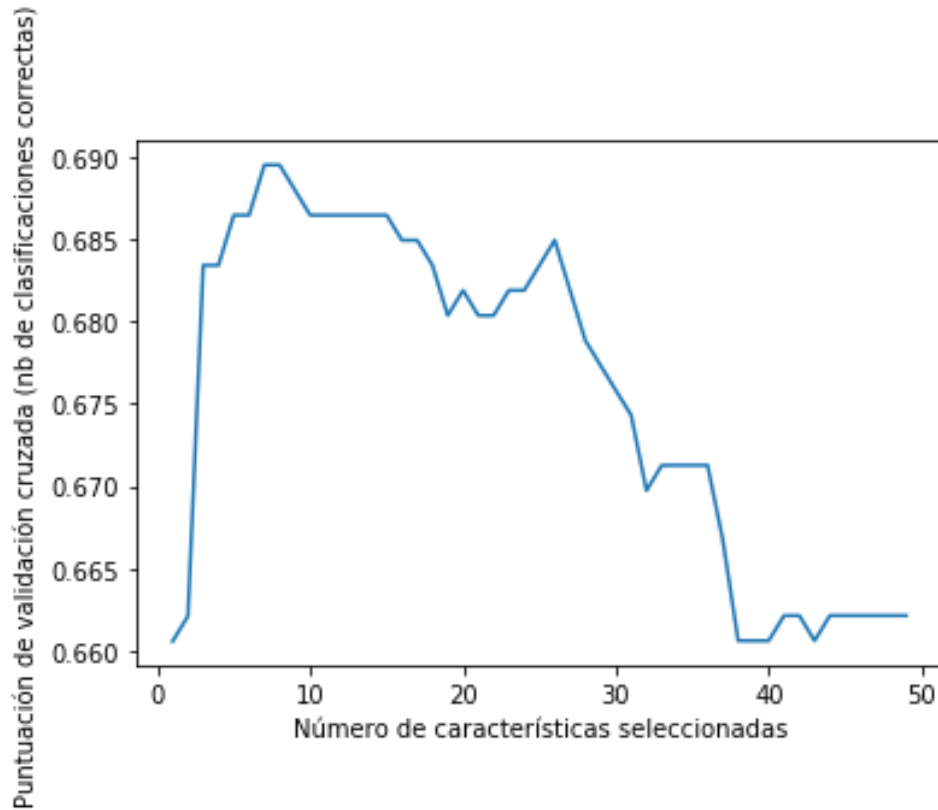


Figura 1.3 Gráfica de selección de variables

```

valores iniciales X (657, 49)
valores iniciales Y (657,)
SELECCION DE VARIABLES [ True False False False False False False True False False False
False False False False False False False False False False False False
False True False False True False False True False False False False
False False False False True True True True False False False True
True]

```

Figura 1.4 Matriz de selección de variables

Regresión logística

La regresión logística ha sido históricamente una herramienta bastante importante y útil para el análisis de datos en investigación clínica y epidemiología. La regresión logística permite discriminar entre dos clases, en términos de un conjunto de variables numéricas, en el papel de predictores [2]. Los objetivos principales de un modelo de regresión logística son:

- Obtener una estimación no sesgada o ajustada de la relación entre la variable dependiente (o resultado) y una variable independiente.

- Evaluar varios factores simultáneamente que estén presumiblemente relacionados de alguna manera (o no) con la variable dependiente.
- Construir un modelo y obtener una ecuación con fines de predicción o cálculo del riesgo.

La regresión logística, a pesar de su nombre, es un modelo lineal para la clasificación en lugar de regresión [2]. La regresión logística es un algoritmo de clasificación lineal ampliamente utilizado en medicina donde una función logística sigmoidea está acoplada a un modelo de regresión lineal. La función utilizada para representar la hipótesis del clasificador está dada por la función sigmoide, cuyos valores de salida son 0 y 1, con una frontera de decisión de 0.5. Los resultados pueden interpretarse como la probabilidad de que la entrada pertenezca a la clase positiva, $P(y = 1; |x; \theta)$, o la negativa $P(x; \theta)$. Donde los resultados se encuentran en el intervalo (0,1), como se muestra en Ecuación 1.3.

$$P(x; \theta) + P(x; \theta) = 1$$

Ecuación 1.3

Despejando la Ecuación 1.3 se obtiene la Ecuación 1.4 que es la probabilidad de que la entrada pertenezca a la clase negativa.

$$P(x; \theta) = 1 - P(x; \theta)$$

Ecuación 1.4

Cuando la probabilidad de que y sea 1 es mayor de 0.5 entonces el clasificador predice 1. Cuando esta probabilidad es inferior a 0.5 predice 0. Para realizar las predicciones el algoritmo utiliza la función sigmoide que se puede ver en la Figura 1.5. La función sigmoide, $h_{\theta}(x)$, se basa en unos parámetros θ que son desconocidos y los valores de cada set de entrenamiento x_0, x_1, x_3, x_n . En la Ecuación 1.5 se muestra la fórmula función sigmoide, donde $-\theta^t x$ es el vector de parámetros o pesos de cada una de las variables predictoras.

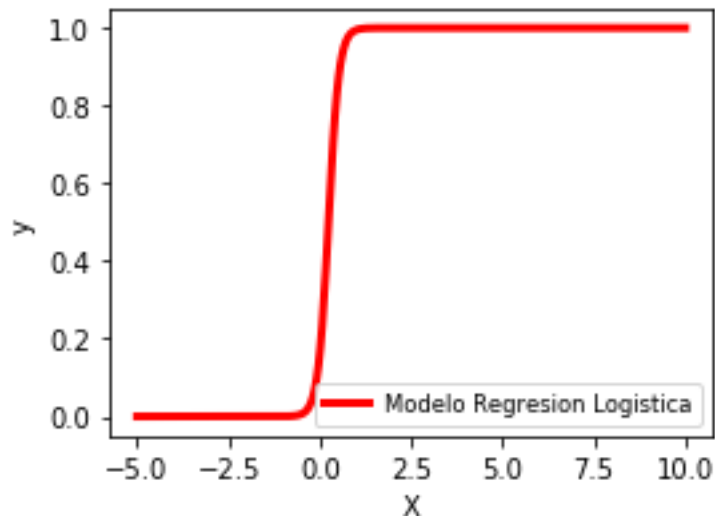


Figura 1.5 Función Sigmoide

$$h_{\theta}(x) = \frac{1}{1 + e^{-\theta x}}$$

Ecuación 1.5

Para encontrar la frontera de decisión en problemas que no son linealmente separables se utiliza la función de costos de mínimos cuadrados, esta permite ajustar los parámetros o los θ y lograr encontrar una función que clasifique entre dos clases, en la Ecuación 1.6 se muestra la función de costos.

$$J(\theta) = \frac{1}{N} \sum_{i=1}^N \frac{1}{2} (h_{\theta}(x^i) - y^i)^2$$

Ecuación 1.6

Donde $h_{\theta}(x^{(i)})$ corresponde a la predicción del clasificador y $y^{(i)}$ es la respuesta esperada, a la mitad del cuadrado de la diferencia entre ellas se le llama función de error cuadrático medio, que en adelante se escribirá como $\text{cos}()$ como se muestra en la Ecuación 1.7.

$$\text{cos}(h_{\theta}(x^i), y^i) = \frac{1}{N} (h_{\theta}(x^i) - y^i)^2$$

Ecuación 1.7

Para lograr la predicción correcta mediante el uso de la función de $\cos()$ se hace necesario aplicar el Gradiente Descendente como se muestra en la Ecuación 1.8, este permite encontrar en una función convexa un punto mínimo local óptimo en el plano.

$$\cos(h_{\theta}(x), y) = \begin{cases} -\log(h_{\theta}(x)), & \text{si } y = 1 \\ -\log(1 - h_{\theta}(x)), & \text{si } y = 0 \end{cases}$$

Ecuación 1.8

Lo anterior se puede expresar en una línea como se indica en la Ecuación 1.9:

$$\cos(h_{\theta}(x^i), y^i) = -y \log(h_{\theta}(x)) - (1 - y) \log(1 - h_{\theta}(x))$$

Ecuación 1.9

Reemplazando la Ecuación 1.7 en la Ecuación 1.6 se consigue la Ecuación 1.10

$$J(\theta) = \frac{1}{N} \sum_{i=1}^N \cos(h_{\theta}(x^i), y^i)$$

Ecuación 1.10

Finalmente compactando las Ecuaciones 1.10 y 1.9 se consigue la función de costos con gradiente descendente expresada en la Ecuación 1.11.

$$J(\theta) = \frac{1}{N} \left[\sum_{i=1}^N -y^i \log h_{\theta}(x^i) + (1 - y^i) \log (1 - h_{\theta}(x^i)) \right]$$

Ecuación 1.11

Hasta ahora se tiene una función de costos que permite encontrar los valores de los θ para la predicción correcta en un conjunto de datos determinado. Esta función presenta un problema de optimización llamado sobre entrenamiento (overfitting), que es el aprendizaje excesivo sobre el set de entrenamiento (memorización) este evita que se consiga un modelo generalizado que pueda funcionar para cualquier set de datos y no solo en el set de datos que se entrenó. Para resolver este problema de optimización se minimiza la función de costos y se le agrega un factor de regularización que permite controlar la complejidad del modelo. La regularización consiste en reducir la importancia de los parámetros θ viéndose modificada la función de costos por la adición de la sumatoria de todos los parámetros θ con un factor llamado parámetro de regularización, λ . Obteniendo como resultado la Ecuación 1.12 [3].

$$\min - \frac{1}{N} \sum_{i=0}^N [Y_n \log h_{\theta}(x) + (1 - Y_n) \log (1 - h_{\theta}(x))] + \lambda \|\theta\|^2$$

Ecuación 1.12

Sabiendo que N es el número de variables, θ son los parámetros de cada variable, y es el vector de respuesta (solo maneja valores binarios (0,1)), y λ es el parámetro de la regularización. Es posible aumentar las capacidades de clasificador mediante la aplicación de transformaciones poligonales a las entradas. En cuyo caso, el límite de decisión puede ser no lineal y problemas más difíciles de manejar. En este modelo, las probabilidades que describen los posibles resultados de un único ensayo se modelan mediante una función logística.

Máquinas de soporte vectorial

Las máquinas de soporte vectorial en inglés Support Vector Machine (SVM) son uno de los métodos de aprendizaje supervisado para problemas de clasificación de dos clases. SVMs tratan problemas linealmente no separables y buscan separar los datos con una gran brecha o hiperplano. En la Figura 1.6 se muestra la línea punteada que indica la frontera de decisión, la distancia que existe entre las dos líneas punteadas se llama margen, los puntos que caen sobre la frontera de decisión se llaman vectores de soporte. En el ejemplo se puede ver tres puntos (dos ejemplos positivos y uno negativos) que se encuentran en la frontera de decisión.

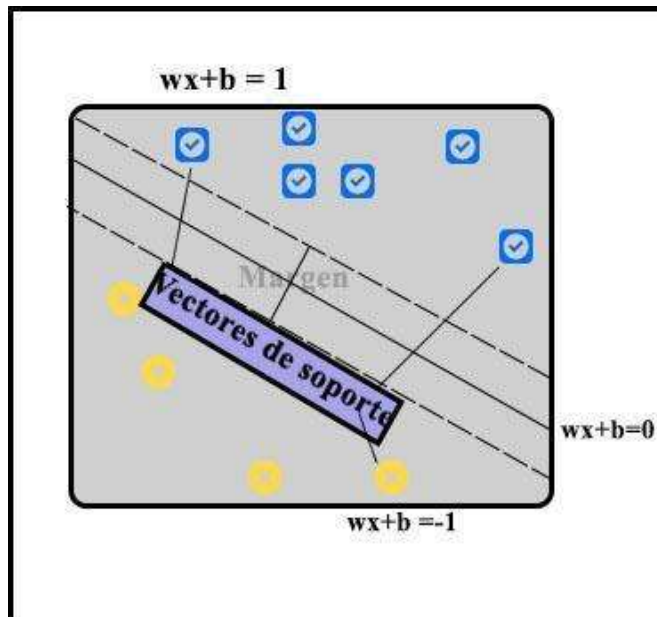


Figura 1.6 Hiperplano separación óptimo

La función de costo de un SVM busca maximizar el margen o distancia entre la frontera de decisión y los puntos que se desea separar, de esta forma se puede obtener el hiperplano de máximo margen. El hiperplano (frontera) de decisión se define mediante la Ecuación 1.12 [2].

$$W^T X + B = 0$$

Ecuación 1.12

Los hiperplanos que caen sobre el margen se muestran en las Ecuaciones 1.13 y 1.14.

$$W^T X + B = 1$$

Ecuación 1.13

$$W^T X + B = -1$$

Ecuación 1.14

Teniendo en cuenta que el vector de pesos W es perpendicular al plano correspondiente, se define $X^+ = X^- + \lambda w$ como la relación que existe entre del margen M con el vector w para encontrar el hiperplano que maximiza el margen como se muestra en la Figura 1.7.

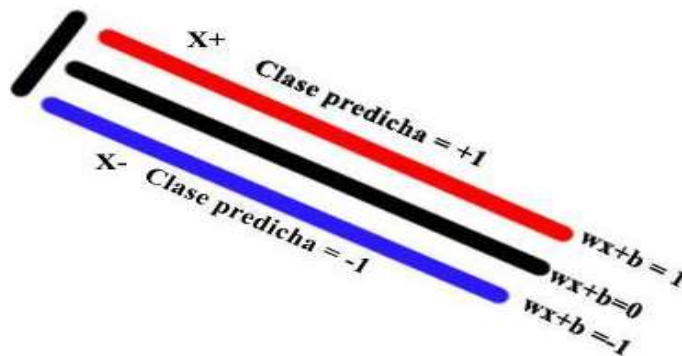


Figura 1.7 Márgenes del hiperplano

El objetivo de un SVM es encontrar el valor de w que maximice el margen M , conociendo que el margen se define mediante la Ecuación 1.15.

$$M = \frac{2}{\|w\|}$$

Ecuación 1.15

El valor del vector w que maximiza el margen M está dado por las Ecuaciones 1.16, 1.17 y 1.18.

$$\min \frac{1}{2} \|w\|^2$$

Ecuación 1.16

sujeto a

$$y_i(\vec{X}_i \cdot w + b) \geq 1$$

Ecuación 1.17

La anterior formulación asume que los datos son linealmente separables. Cuando no lo son, no es suficiente con maximizar el margen, también se minimizar los errores de clasificación convirtiéndose en un problema de optimización que se soluciona mediante las Ecuaciones 1.18 y 1.19, donde Los ξ representan variables de holgura, que relajan las restricciones y aportan al objetivo y donde C es una constante lo suficientemente grande, elegida por el usuario, que permite controlar en qué grado influye el término del coste de ejemplos no-separables en la minimización de la norma, es decir, permitirá regular el compromiso entre el grado de sobreajuste del clasificador final y la proporción del número de ejemplos no separables, así, un valor de C muy grande permitiría valores de ξ_i muy pequeños .

$$\min \xi \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i$$

Ecuación 1.18

sujeto a,

$$y(\vec{X}_i \cdot w + b) \geq 1 - \xi$$

Ecuación 1.19

Algunas de ventajas de las máquinas de vectores de soporte son:

- Efectiva en espacios de alta dimensión.

- Utiliza un subconjunto de puntos de entrenamiento en la función de decisión (llamados vectores de soporte), utilizando eficientemente la memoria.
- Versátil: diferentes funciones del núcleo pueden ser especificados para la función de decisión. Núcleos comunes se proporcionan, pero también es posible especificar núcleos personalizados.

Para el caso de los problemas que no son linealmente separables se recomienda la inclusión de las funciones núcleo o kernel tiene como efecto un mapeo de las entradas a un espacio de alta dimensionalidad, donde los datos si serán linealmente separables.

La función del núcleo tiene como objetivo separar los vectores de soporte del resto de los datos de entrenamiento, este es un problema de programación cuadrática (QP).

En este proyecto se usa la función núcleo RBF (Radial Base Function) También es conocido como el núcleo "exponencial". La funciones de kernel RBF toman la forma $\exp(-\gamma|x - x_0|^2) \cdot \gamma$. Donde γ es una constante de proporcionalidad cuyo rango de valores útiles debe ser estimado para cada aplicación en particular. Otras funciones kernel comúnmente usadas son: linear, polynomial, sigmoide.

Este núcleo es infinitamente diferenciable, lo que implica que GPs con este kernel como función de covarianza tienen las derivadas cuadradas medias de todos los órdenes, y por lo tanto son muy suaves.

1.3.2 Marco conceptual

Regularización

Se le llama regularización al proceso que controla la complejidad de un modelo, resolviendo los problemas de optimización reduciendo la importancia de los parámetros de θ en las funciones del modelo.

La regularización tiene como objetivo realizar un intercambio apropiado entre la fiabilidad de los datos de entrenamiento y las bondades del modelo. En procedimientos de aprendizaje supervisado, el intercambio se realiza a través de la minimización del riesgo total.

Regularización es cuando añades un término que suaviza los resultados para evitar el sobreajuste (overfitting) [19].

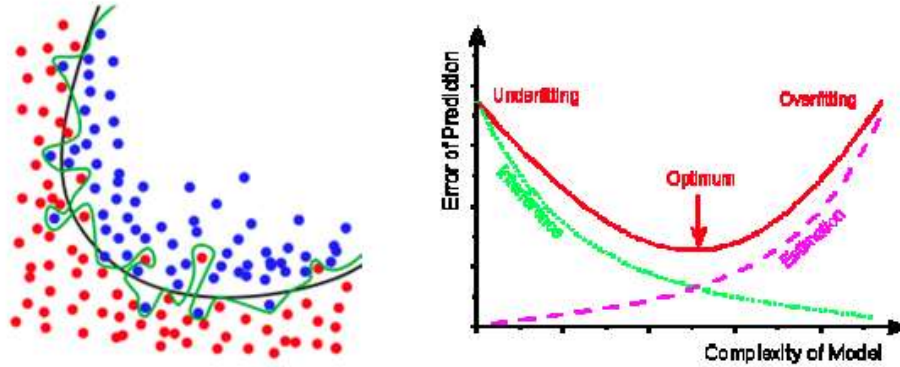


Figura 1.8 Sobreajuste Overfitting

Esto quiere decir que, cuando entrenas una máquina, la máquina intentará adaptarse lo máximo posible a los datos de entrenamiento, ya que quiere evitar lo más posible el error. Pero en machine learning esto no es óptimo ya que un sobreajuste en los datos de entrenamiento puede producir una poca capacidad de predicción en los datos reales.

Generalización

Dentro de la inteligencia artificial existen un concepto que se le llama generación y está dependiente de un entrenamiento de aprendizaje supervisado y el éxito de predecir con precisión se le llama generación, es lo contrario a error.

Generalización: ¿Cómo conseguir que la red funcione bien con datos distintos a los del conjunto de entrenamiento?

Como si se tratase de un ser humano, las máquinas de aprendizaje deberán ser capaces de generalizar conceptos. Visualice que mira un perro Labrador por primera vez en la vida y le dicen “eso es un perro”. Luego le enseñan un Caniche y le preguntan: ¿eso es un perro? Dirá “No”, pues no se parece en nada a lo que aprendió anteriormente. Ahora piense que su tutor le enseña un libro con fotos de 10 razas de perros distintas. Cuando vea una raza de perro que desconoce seguramente será capaces de reconocer al cuadrúpedo canino al tiempo de poder discernir qué es un gato no es un perro [20].

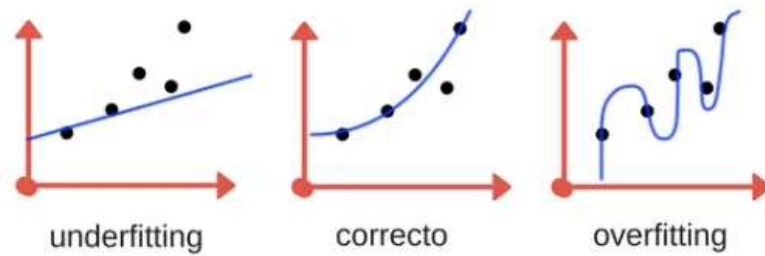


Figura 1.9 Equilibrio del Aprendizaje

Siempre que se crea una máquina de aprendizaje se debe tener en cuenta que pueden caer en uno de estos problemas por no poder generalizar correctamente el conocimiento. Underfitting indicará la imposibilidad de identificar o de obtener resultados correctos por carecer de suficientes muestras de entrenamiento o un entrenamiento muy pobre. Overfitting indicará un aprendizaje “excesivo” del conjunto de datos de entrenamiento haciendo que nuestro modelo únicamente pueda producir unos resultados singulares y con la imposibilidad de comprender nuevos datos de entrada.

Matriz de confusión

Una matriz de confusión es una técnica para resumir el rendimiento de un algoritmo de clasificación. La precisión de la clasificación por sí sola puede ser engañosa si tiene una cantidad desigual de observaciones en cada clase o si tiene más de dos clases en su conjunto de datos [14].

El cálculo de una matriz de confusión puede darle una mejor idea de lo que su modelo de clasificación está haciendo bien y qué tipo de errores está cometiendo.

La matriz de confusión muestra las formas en que su modelo de clasificación se confunde cuando hace predicciones.

Le da una idea no solo de los errores cometidos por su clasificador, sino más importante de los tipos de errores que se están cometiendo.

Es este colapso lo que supera la limitación de usar solo la precisión de clasificación. Es una matriz que muestra las clasificaciones prevista y las reales. Una matriz de confusión es de tamaño $L \times L$, donde L es el número de diferentes valores de la etiqueta.

Predicho	/	Positivo	Negativo
----------	---	----------	----------

Real		
Positivo	VP	FP
Negativo	FN	VN

Tabla 1 Matriz de confusión

. Donde,

V P: Verdaderos Positivos

V N: Verdaderos Negativos

F P: Falsos Positivos

F N: Falsos Negativos.

Los resultados del clasificador se pueden interpretar mediante el uso de la gráfica ROC como se muestra en la Figura 1.8.

Curva ROC

En la teoría de detección de señales, una curva ROC (acrónimo de Receiver Operación Característica, o Característica Operativa del Receptor) es una representación gráfica de la sensibilidad frente a la especificidad para un sistema clasificador binario según se varia el umbral de discriminación. Otra interpretación de este gráfico es la representación de la razón o ratio de verdaderos positivos frente a la razón o ratio de falsos positivos [7].

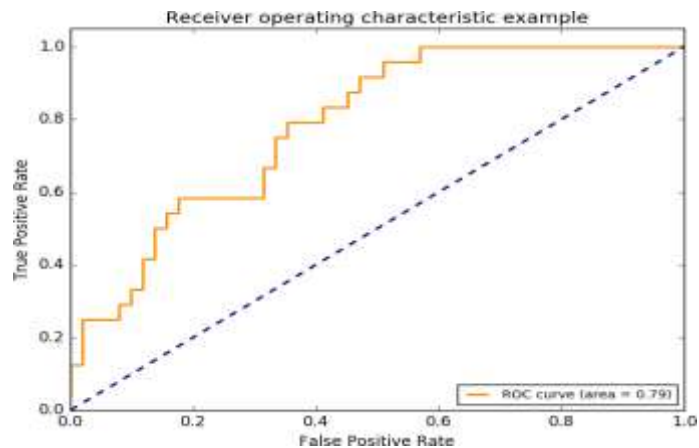


Figura 1.10 Gráfica de ROC

Sensibilidad (Recall): también llamada razón de Verdaderos Positivos (VPR). La sensibilidad indica la capacidad del clasificador para dar como casos positivos los casos

realmente positivos (enfermos), útil para detectar pacientes enfermos. La sensibilidad se define mediante la ecuación 1.20.

$$VPR = \frac{VP}{(VP + FN)}$$

Ecuación 1.20

Precisión (tasa de error): también llamada accuracy en inglés, corresponde a la tasa de predicciones correctas hechas por el modelo sobre un conjunto de datos. La precisión puede verse como una medida de la exactitud o la calidad, mientras que la sensibilidad es una medida de la integridad. La precisión se suele calcular mediante el uso de un conjunto de pruebas independientes que no fueron utilizada durante el proceso de aprendizaje. Sin embargo, para casos en los que el conjunto de datos es pequeño, porque el número de casos es reducido, otras técnicas de estimación de precisión más complejas son utilizadas tales como validación cruzada y rutina de caras. La formulación matemática para esta medida está dada por la Ecuación 1.21.

$$precision = \frac{(VP)}{(VP + FP)}$$

Ecuación 1.21

Especificidad: también llamada razón de verdaderos negativos. Indica la capacidad del clasificador para dar como casos negativos los casos realmente negativos (sanos). Útil para detectar pacientes sanos. La especificidad se define mediante la Ecuación 1.22.

$$FPR = \frac{VN}{(FP + VN)}$$

Ecuación 1.22

1.4 METODOLOGÍA

1.4.1 Línea de investigación

La Universidad del Sinú seccional Cartagena cuenta con la línea de investigación de inteligencia artificial, el cual incluye temática específica para el desarrollo de estudios de datos a través del aprendizaje automático.

Para la realización de predicciones a través de algoritmos con el lenguaje de programación Python se realizarán diferentes técnicas y métodos para hacer una comparación de cuáles son las mejores técnicas y además señalar las características más importantes para el uso de machine learning para la toma de decisiones.

1.4.2 Tipo de investigación

La investigación a utilizar es de tipo aplicada en cual se busca hacer una descripción de una comparación varios análisis de modelos que se van aplicar en las distintas técnicas de aprendizaje automático para el desarrollo de un modelo predictivo de casos de hipertensión en la ciudad de Cartagena. El propósito de la investigación utilizada consiste en describir situaciones, eventos o cualquier fenómeno que someta análisis como es la ciencia de datos y algoritmos que miden o evalúan diversos aspectos para obtener predicciones que puedan ayudar a la toma de decisiones por medio de matemáticas y estadísticas.

1.4.3 Metodología descriptiva

La metodología para el desarrollo de este proyecto es una estructura que se maneja frecuentemente para explicar de cierta manera descriptiva las investigaciones, en donde es posible aplicar los resultados. En la programación de algoritmos de aprendizaje automático con el lenguaje Python, se utilizó el trazado de objetivos para el desarrollo de este, el cual consiste en realizar diferentes técnicas o métodos para analizar datos y hacer predicciones a través de funciones matemáticas y estadísticas. Cada una de las tareas fueron ejecutadas así secuencialmente hasta llegar el objetivo y cumplir con el alcance. En otras palabras, esta metodología también es conocido como modelo cascada, caracterizado por utilizar procesos secuenciales para el desarrollo, diseño, implementación, pruebas o validaciones, integración y mantenimiento.

Teniendo en cuenta la enorme y creciente necesidad de la medicina para tomar decisiones correctas y oportunas de los pacientes, se propone desde el campo de la ingeniería algunas soluciones tecnológicas que generen el apoyo al profesional durante el proceso de atención de pacientes.

Este proyecto de investigación plantea la comparación de dos modelos de predicción de riesgos para trastornos hipertensivos. Así mismo, se plantea implementar técnicas de machine learning y deep learning para realizar predicción del riesgo que puedan tener los pacientes de padecer trastornos hipertensivos como ya se ha venido utilizando en otras patologías para la predicción de complicaciones e interacciones con el tratamiento [13].

La presente investigación tendrá como fundamento la predicción mediante la lectura de variables clínicas o signos desde sensores que pueden medir, por ejemplo: la presión de la sangre, los niveles de azúcar en la sangre, niveles de oxígeno y algunas condiciones de temperatura y el medio ambiente en las actividades diarias del paciente. Así mismo, se construirá la base de datos de los pacientes utilizando herramientas que permitan la recolección y manipulación de la información de manera sencilla. Se implementarán técnicas de análisis de variables para determinar cuáles son las que más relevancia e incidencia tiene respecto a la variable de respuesta, esto se realizará por cada algoritmo. La selección de variables permite especificar cómo se introducen las variables independientes en el análisis sobre el conjunto de datos.

Alcance

Funciones	Características
Selección de variables usando técnica de feature selection	Las principales razones para usar la selección de funciones son: <ul style="list-style-type: none">• Permite al algoritmo de aprendizaje automático entrenar más rápido.• Reduce la complejidad de un modelo y lo hace más fácil de interpretar.• Mejora la precisión de un modelo si se elige el subconjunto correcto.• Reduce el sobreajuste.
Implementación de algoritmo de máquina de soporte vectorial en python	Se requiere implementar un algoritmo de máquina de soporte vectorial en python utilizando la librería scikit-learn , sklearn
Implementación de algoritmo de regresión logística en Python	Se requiere implementar un algoritmo de regresión Logística de la librería Sklearn de python.
Comparación de algoritmos	aplicar las técnicas machine learning para comparar los dos modelos en cuanto a las mediciones que evalúan cada modelo y señalar cuál de los modelos comparados es mejor para predicciones de casos de hipertensión en la ciudad de Cartagena

2 ANÁLISIS DEL PROBLEMA

2.1 REQUISITOS ESPECÍFICOS IEEE-830

Requerimientos Funcionales

Identificación del requerimiento: RF01

Nombre del Requerimiento: Set de datos

Características: Un set de datos que esté en un archivo csv que esté separado por comas y clasificado..

Descripción del requerimiento:	Es necesario para la realización de los algoritmos.
Requerimiento NO funcional:	
Prioridad del requerimiento:	Alta

Tabla 2 Requerimiento - RF01 Set de datos

Requerimientos No Funcionales IEEE-830

Identificación del requerimiento:	del RNF01
Nombre del Requerimiento:	Entorno de desarrollo con jupyter notebook con código Python y las librerías Scikit-learn y pandas..
Características:	Ambiente de desarrollo, variables de entorno que normalmente es posible manejar con solo Python por cuestión de practicidad para el desarrollo Los algoritmos se recomienda esta manera.
Descripción del requerimiento:	Para la ejecución de los algoritmos es necesario tener todos los requisitos y permisos del sistema.
Prioridad del requerimiento:	Alta

Tabla 3 Requerimiento -RNF01 Entorno

2.2 REQUISITOS FUNCIONALES

2.2.1 Algoritmos

Los algoritmos deben ser capaz de procesar el set de datos y segmentarlo o dividirlo de tal manera que sea un 80% entrenamiento y 20% para pruebas y validaciones.

Las técnicas implementadas deben ser capaz de predecir si un paciente es capaz de estar en riesgo de hipertensión.

2.3 REQUISITOS NO FUNCIONALES

2.3.1 ENTORNO DESARROLLO

El entorno donde estará desarrollado las técnicas y métodos requiere de librerías de Python específicos para la ejecución funcional.

Las técnicas utilizadas deben tener buenas prácticas para que sirva de ejemplo para investigadores y estudiantes que necesiten documentación de algoritmos de aprendizaje automático con el lenguaje de programación Python.

3 DISEÑO DE LA SOLUCIÓN

3.1 Organización de la información

3.1.1 Recolectar la información de los datos que se necesitan

Es importante centralizar y dejar una idea clara del objetivo de la implementación de algoritmos de aprendizaje automático que se van a utilizar para también tener que investigar el tipo de datos que se necesitan recolectar para resolver problemas con las técnicas que se van a probar. Es importante la cantidad y la calidad de información ya que esta va relacionada de alguna forma en la calidad de clasificación de los algoritmos de predicción.

3.1.2 Preparación del set de datos.

Para este punto se unificará toda la información recolectada para el análisis de las variables que se presentaron y el número total de muestras para la investigación. El número correcto de datos necesarios para tener una buena precisión está directamente ligado a que los datos estén relacionados, tengan una cantidad balanceada para cada resultado que se busca.

Una vez obtenido todo en cuenta se define el porcentaje de información para cada grupo el cual consisten en dos partes, una de ellas encargada para el entrenamiento y el otro set de datos es para validar el modelo.

3.2 Preparación del entorno de desarrollo.

3.2.1 Instalación de Anaconda.

Se requiere de un entorno donde se pueda sacar el provecho a entornos que están preparados para el desarrollo de machine learning con el lenguaje de programación Python.

Por la manera en cómo se va desarrollar el proyecto de manera aplicada es necesario un entorno que sea de base para el uso de implementación de algoritmos con aprendizaje automático profundo.

Por selección libre se escogió un entorno que es muy intuitivo y dinámico para realizar pruebas e investigaciones con machine learning y otros propósitos, se instaló la

plataforma para data Science, llamada Anaconda que sirve además para sacarle provecho a la implementación y el trabajo colaborativo y también se puede considerar como manejador de dependencias o paquetes de librerías para trabajar con Python.

3.2.2 Instalación de librerías.

En la instalación de Anaconda la mayoría de las librerías están descargadas y pre-instaladas por defecto para hacer algoritmos básicos con machine learning, pero también es posible que se necesiten otras que pueden servir de ayuda, como pudo ser pandas que fue agregada para hacer gráficas y diagramas desde jupyter notebook.

Se agregan todas las librerías que se van a utilizar en las técnicas de machine learning, para poder trabajar con todos los requisitos que se requiere para hacer pruebas con datos reales y mostrar los resultados para comparar su respectivo análisis a la respuesta que se evalúen. Así mismo se harán diferentes pruebas implementado las librerías que se van a usar para la definición de modelos y casos donde se aplique para el aprendizaje automático.

3.3 Implementación de técnicas de IA.

3.3.1 Análisis de las técnicas.

Se analiza el set de datos que se va a trabajar y si es necesario aplicar técnicas que ayuden a identificar de un set de datos recolectado las variables importantes que tendrán mayor peso y que ayudan de alguna forma a mejorar los modelos en cuanto a precisión y clasificaciones se aplica la técnica elegida.

Se tiene que empezar a modelar los diferentes algoritmos y técnicas para la prueba de los datos recolectados. Se realizan los algoritmos donde se implementa el dataset construido y normalizado luego se hacen las primeras pruebas con la información y los modelos que se van creando a su medida.

3.3.2 Desarrollo de algoritmo de selección de variables

Para el análisis de la data que se va utilizar es necesario tener los datos totalmente normalizados para que la información que se va trabajar en los algoritmos estén mejor segmentados por las variables más importante hasta quitar las que no son necesarias, esta selección de variables también se hace para que no se desarrollen sobre ajustes en los modelos de clasificación.

3.3.3 Normalización del set de datos

De la colección de información suministrada primero se organizó el set de datos en un archivo CSV delimitado por coma para la lectura de datos en los métodos o técnicas de machine learning que se implementarán para la comparación de los diferentes modelos.

Las columnas y filas fueron revisadas para validar que la información esté uniforme y organizada para que la lectura e implementación de los algoritmos de esa forma evitar tener algún problema de lectura de la información.

3.4 Desarrollo de algoritmos

3.4.1 Desarrollo de algoritmo de regresión logística.

se implementa el algoritmo de regresión logística por su característica de clasificador lineal y así luego hacer una comparación con otra técnica y evaluar el comportamiento del aprendizaje automático visto desde los aspectos de precisión, sensibilidad y especificidad, sabiendo utilizar un set de datos para detectar patrón para un análisis de variables seleccionadas para la clasificación siendo estas a su vez las que sean más relevantes para la predicción o clasificación en el área de la salud.

3.4.2 Desarrollo de algoritmo de máquinas de soporte vectorial

Se desarrolla el algoritmo que complementa las máquinas de soporte vectorial con el lenguaje Python para la comparación de algunos aspectos importantes en cuanto sensibilidad, precisión y especificidad. Utilizando las librerías que anaconda proporciona para el uso de inteligencia artificial con machine learning y algoritmo de aprendizaje supervisado.

3.5 Comparación de los modelos

3.5.1 Análisis de resultados

Se desea saber cuál de los modelos es el más adecuado para la predicción de pacientes en riesgo de hipertensión, así como saber las ventajas y desventajas de cada uno de los modelos utilizados. Se compara cada uno de los modelos y se elige uno como mejor para la predicción de diferentes pacientes con riesgo en hipertensión.

3.5.2 Tabla comparativa de los resultados obtenidos

Se organiza la información obtenida con los modelos que se entrenaron de por lo menos los dos más útiles para el aprendizaje de automático como hacer predicción con datos históricos de pacientes con riesgos de hipertensión B1, además una descripción breve de las ventajas y desventajas de los modelos utilizados para la clasificación o predicción

4 DESARROLLO

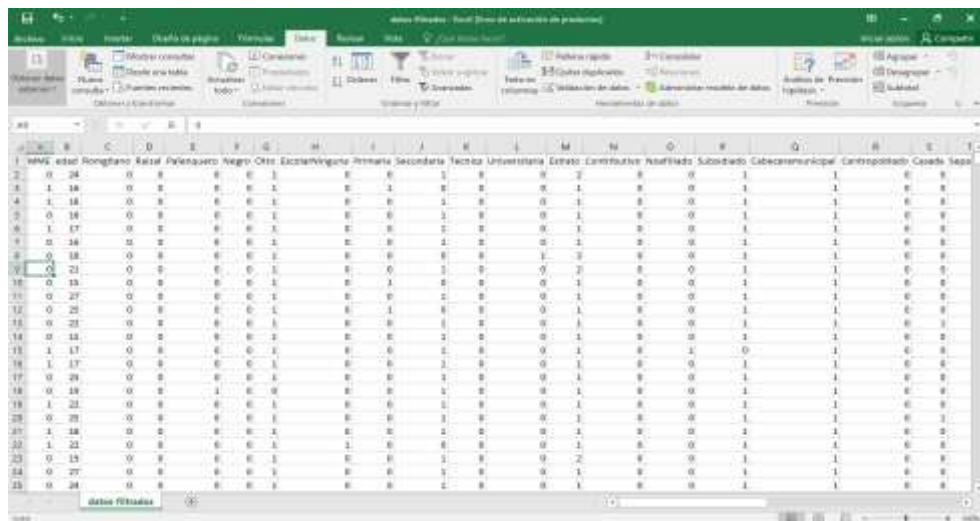
4.1 Colecta y preparación del set de datos.

El funcionamiento de los algoritmos y modelos de Machine Learning necesitan tener un conjunto de datos que esté claramente implicados a la utilización para la base de los todos los modelos que va a entrenar a través de diferentes técnicas para obtener clasificaciones que ayuden a predecir mejor, siendo esta parte importante para la construcción de los modelos.

En el trabajo de recolectar y organizar los datos se realizaron en dos partes el cual la primera fue recolectar todos los datos unificarlos en un solo documento y la segunda es prepararlos en el formato que se va trabajar con los algoritmos.

4.1.1 Colecta de datos

En la búsqueda de los datos contamos con la ayuda del docente disciplinar para encontrar un contacto que permitió acceder a una información de una investigación reciente a pacientes con posibles casos de hipertensión, está ya se encontraba preparada en dos archivos organizados por características y resultados. El trabajo fue unificar los dos archivos en uno solo y empezar analizar los datos para que pudieran ser tratados con el mismo orden y número de filas y columnas.



The image shows a screenshot of a Microsoft Excel spreadsheet. The spreadsheet has a grid with columns labeled A through S and rows numbered 1 through 25. The data in the spreadsheet consists of numerical values, likely representing counts or frequencies, organized into columns. The first column (A) contains values ranging from 0 to 1. The other columns (B through S) contain values ranging from 0 to 1. The spreadsheet is titled 'datos filtrados' and has a green header bar with the Microsoft Office logo and the text 'datos filtrados - libro1.xlsx de archivos de programas'.

Figura 4.1 Set de datos

4.1.2 Preparación del formato

Para las técnicas de manejo de data con Machine Learning ya se han venido haciendo trabajos que incluyen librerías para el manejo, estas requieren de formatos y de estructuras que se van aplicar en nuestros modelos.

Luego de haber recolectado la información y organiza en una sola parte, se cambia al formato el cual se va aplicar en los algoritmos. El archivo tiene que ser formato CSV en el cual los datos deben ir en una columna separados por coma.

Para ello se utilizó una función muy sencilla en Excel, el cual fue “CONCATENAR ()”, la función recibe las columnas que se van a unir, pero esta se le tuvo que añadir una concatenación extra el cual era colocar entre el medio de las columnas vecinas el separador coma, para poder cumplir con el formato que se va implementar.

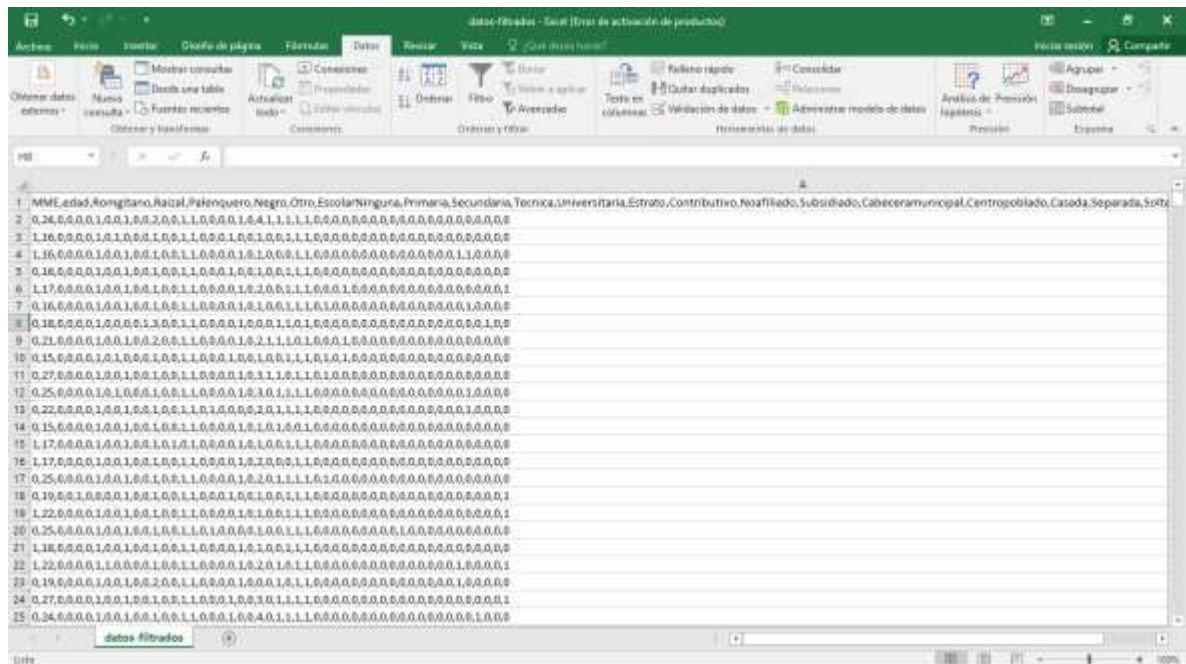


Figura 4.2 Formato set de datos

4.1.3 Técnica de selección de variable

Por motivos de mejorar el set de datos el cual fue revisado minuciosamente para que tuviera igualdad de parámetros de características y además que fueran de utilidad para para los modelos que se van a implementar, de todo eso salieron un total de 50 columnas.

Normalmente se puede trabajar con cualquier número de columnas que equivalen al mismo número de características, pero esto no todas las veces es bueno para algunos

modelos, entonces es cuando se analizan las técnicas de selección de variables, el cual su objetivo es mejorar el set de datos, esté de acuerdo a la configuración y método selecciona las variables que le ayudan a mejorar la precisión para la decisión categóricas de los modelos entrenados.

La técnica de selección de variable que se van a usar, devolverá un subconjunto de datos del original con las características más óptimas según los criterios que se evalúen. La técnica es ofrecida por sklearn el nombre de la librería es Feature Selection, el cual su implementación es válida para la mejora de los tiempos de entrenamientos cuando los datos son muchas variables para la clasificación y además también hace tener mejor visualización de los datos que se trabajan y se van a seleccionar.

Se ha hecho en base a cómo controlar los problemas de desbalance de datos referentes a las clases para evitar problemas de clasificación, de manera general la técnica selecciona las clases con mayor proceso de clasificación e ignora las clases minoritarias.

4.2 Preparación del entorno

4.2.1 Instalación de Anaconda

Por sugerencias de otros desarrolladores de inteligencia artificial que han trabajado, recomiendan el uso de anaconda para el desarrollo y análisis enfocada a los algoritmos de machine learning que estudian los datos. Sabiendo que para el desarrollo es necesario trabajar con bastantes librerías que son requeridas y que esta plataforma libre ofrece todo el paquete básico para el desarrollo de investigaciones para dos lenguajes R y Python.

Esta se consigue en la web y es totalmente gratis, además es multiplataforma, en este caso trabajo los mismos algoritmos en Windows y Mac IOS. En ambos sistemas operativos funcionan sin problema. Su instalación es rápida en Windows y en Mac el proceso es algo diferente, una vez descargado se ejecuta y se hace la secuencia de pasos.

Cabe resaltar que tiene diferentes formas de instalarse, en Windows normalmente las aplicaciones tienen un patrón de instalación y en Mac también cuenta con una aplicación para instalación rápida y que en esta ocasión es parecida en pasos para ambos SO.



Figura 4.3 instalador Anaconda

Una vez instalado anaconda puede verificar que tenga todo instalado, eso se hace iniciando el CMD de la PC y ejecutando los siguientes comandos, que están en la siguiente imagen.

```
C:\WINDOWS\system32\cmd.exe
C:\Users\Ñejo>Anaconda -V
anaconda Command line client (version 1.6.14)
C:\Users\Ñejo>python -V
Python 3.6.5 :: Anaconda, Inc.
C:\Users\Ñejo>conda -V
conda 4.5.4
```

Figura 4.4 Versiones del entorno

Como se puede ver se tiene el entorno preparado para trabajar Machine Learning con Python. Esto no es todo anaconda tiene soporte para la instalación y manejo de librerías, siendo útil para la instalación de las que se van a usar adicionales que no están el paquete predeterminado de Anaconda, también es posible realizar entornos de desarrollo variados en donde pudimos trabajar y probar diferentes versiones de Python como lenguaje principal para la investigación.

Para esta ocasión se trabaja con la versión de Python 3.6.5, el cual es una de las que recomienda Anaconda para trabajar con Machine Learning. También se implementó parte de los algoritmos en Mac en donde se trabajó la versión de Python 2.7 que también es recomendada y soportada por Anaconda. De una también se incluyeron las librerías las cual son requeridas para las pruebas de los modelos.

4.3 Implementación de algoritmos

4.3.1 Importaciones necesarias

Para empezar a mostrar el código fuente es necesario separar lo que se está ejecutando con los resultados, pero con jupyter notebook el cual es un entorno para hacer más intuitivo el desarrollo de investigaciones, permite hacer bloques de código que se pueden ejecutar para su prueba siendo así amigable para estos trabajos.

```
In [2]: 1 #Importamos la libreria con la cual vamos a usar para leer el archivo con las datos a utilizar
2 import csv
3
4 #Importamos la libreria numpy para trabajar con los vectores resultante de las columnas y filas del documento
5 import numpy as np
6
7 #Graficas
8 import matplotlib.pyplot as plt
9 import pandas
10 from sklearn.metrics import roc_curve, auc
11
12 #SELECCION
13 from sklearn.model_selection import StratifiedKFold #Tecnica Selección de variable
14 from sklearn.feature_selection import RFECV, SelectFromModel, SelectKBest, f_regression #Técnicas Selección de variable
15 from sklearn.ensemble import ExtraTreesClassifier #Clasificadores
16 from sklearn.pipeline import Pipeline #Clasificadores
17 from sklearn.ensemble import RandomForestClassifier #Clasificadores
18 from sklearn.preprocessing import MinMaxScaler #normalización
19 from sklearn import linear_model # regresion lineal
20 from sklearn import model_selection # regresion lineal
21 from sklearn.metrics import confusion_matrix # regresion lineal
22 from sklearn.metrics import accuracy_score # regresion lineal
```

Figura 4.5 Importaciones y Librerías Python

4.3.2 Métodos de manejo de data set

Se definen tres métodos comunes que son usados por las técnicas implementadas, utilizando funcionalidades de las librerías Pandas, CSV para leer los archivos .csv, el primer método retorna las cabeceras de la data y el otro retornar toda la data contenida en los archivos.

```
in [21]: 1 archivo = '/Users/Rejo/datos-filtrados.csv'
2
3 # cargar encabezados csv
4 def cargar_archivo_encabezado(filename):
5     data = list()
6     with open(filename, newline='') as f:
7         datos = csv.reader(f)
8         for row in datos:
9             if not row:
10                data.append()
11                data.append(row)
12                break
13 #retornamos la fila numero 0 la cual contiene los nombres de las columnas
14 return data[0]
15
16 # cargar archivo csv
17 def cargar_archivo_csv(filename):
18
19 #usamos el metodo reader de la clase csv para leer el archivo
20 dataset = pd.read_csv('/Users/Rejo/Documents/CSV/datos-filtrados.csv', engine='python')
21
22 return dataset
23
24
25 #obtenemos los datos ya cargados
26 set_datos = cargar_archivo_csv(archivo)
27 encabezados = cargar_archivo_encabezado(archivo)
28
```

Figura 4.6 Métodos de manejo de datos

La variable **set_datos**, **encabezados** y características quedan almacenada en memoria estas contienen retornan los métodos **cargar_archivo_encabezado**, **cargar_archivo_csv** y **obtener_x** respectivamente. El contenido de las variables se puede ver en la siguiente imagen.

```

24 set_datos = cargar_archivo_csv(archivo)
25 encabezados = cargar_archivo_encabezado(archivo)

['MME', 'edad', 'Romgitano', 'Raical', 'Palenquero', 'Negro', 'Otro', 'EscolarKinguna', 'Primaria', 'Secundaria', 'Tecnica',
 'Universitaria', 'Estrato', 'Contributivo', 'Noafiliado', 'Subsidiado', 'CabeceraMunicipal', 'Centropoblado', 'Casada', '
 Separada', 'Soltera', 'Unionlibre', 'Viuda', 'Fariada', 'intergeneracion', 'Multiplicidad', 'micronutrientes', 'Preeclampsia',
 'Sepsis', 'Eclampsia', 'Diabetes', 'TORCHS', 'vias-Urinaras', 'autoimmune', 'M50-M64', 'D10-D36', 'D50-D64', 'E10-E14',
 'E65-E68', 'E70-E90', 'I11-I15', 'J00-J06', 'J40-J47', 'O10-O16', 'O20-O29', 'O30-O48', 'O95-O99', 'P05-P08', 'Q50-Q56', 'Z
 00-Z13']

MME  edad  Romgitano  Raical  Palenquero  Negro  Otro  EscolarKinguna  \
0    0    24          0        0            0    0    1            0
1    1    16          0        0            0    0    1            0
2    1    16          0        0            0    0    1            0
3    0    16          0        0            0    0    1            0
4    1    17          0        0            0    0    1            0
5    0    16          0        0            0    0    1            0
6    0    18          0        0            0    0    1            0
7    0    21          0        0            0    0    1            0
8    0    15          0        0            0    0    1            0
9    0    27          0        0            0    0    1            0
10   0    25          0        0            0    0    1            0
11   0    22          0        0            0    0    1            0
12   0    15          0        0            0    0    1            0
13   1    17          0        0            0    0    1            0
14   1    17          0        0            0    0    1            0
15   0    25          0        0            0    0    1            0
16   0    19          0        0            1    0    0            0
17   1    22          0        0            0    0    1            0
18   0    25          0        0            0    0    1            0
19   1    18          0        0            0    0    1            0
20   1    22          0        0            0    0    1            1
21   0    19          0        0            0    0    1            0

```

Figura 4.7 Encabezados y características de los datos

El set de datos está conformado por 657 filas x 50 columnas en total y de una se evidencia lo útil que son las librerías para agilizar la investigación. A continuación, el siguiente método.

```

In [24]: 1 def obtener_x():
2         dataset = pandas.read_csv(archivo , engine='python' )
3         items = []
4         for i in range(1 , len(encabezados)):
5             if(type(encabezados[i]) != None):
6                 items.insert(i,encabezados[i])
7         return dataset.filter(items=items)
8
9
10 X =obtener_x()
11 Y = set_datos['MME']
12
13 print('valores iniciales X ',X.shape)
14 print('valores iniciales Y ', Y.shape)

valores iniciales X (657, 49)
valores iniciales Y (657,)

```

Figura 4.8 Inicialización de variables globales

Se evidencia que el set de datos ya contiene las cabeceras, características y etiquetas, la cual las carteristas constan de 657 filas y 49 columnas, es decir todas las columnas de características menos la columna de etiquetas, las etiquetas hacen una matriz unidimensional con el mismo número de filas que las características.

4.3.3 Técnicas de Machine Learning

Para efectividad del set de datos para los modelos se aplica una técnica de normalización para los datos transformando las características individualmente al escalar de cada característica a un rango dado, pro defecto es 0 a 1, la fórmula que aplica la normalización escalar es:

$$X_{std} = (X - X.min(axis = 0)) / (X.max(axis = 0) - X.min(axis = 0))$$

$$X_{scaled} = X_{std} * (max - min) + min$$

En **sklearn** ya se encuentra una librería que ayuda con la normalización llamada **MinMaxScaler** y es de gran utilidad para avanzar con rapidez:

```
In [27]: 1 #normalizacion de los datos
          2 df = obtener_x()
          3 scaler = MinMaxScaler()
          4 scaled_df = scaler.fit_transform(df)

MinMaxScaler(copy=True, feature_range=(0, 1))
[[0.34375 0.    0.    ... 0.    0.    0.    ]
 [0.09375 0.    0.    ... 0.    0.    0.    ]
 [0.09375 0.    0.    ... 0.    0.    0.    ]
 ...
 [0.09375 0.    0.    ... 0.    0.    0.    ]
 [0.125   0.    0.    ... 0.    0.    0.    ]
 [0.125   0.    0.    ... 0.    0.    0.    ]]
```

Figura 4.9 Normalización de los datos

Se aplicaron técnicas de normalización a todo el set de datos con **DataFrame** y **Numpy** clases de la librería sklearn para el manejo de datos en Python con estas clases, ya se puede entrenar con la data que se ha preparado los modelos que se van a probar.

Encabezado de características y Set de datos tipo DataFrame y Array Numpy tipo float

```
In [30]: 1 #obteniendo columnas
2 columns = cargar_archivo_encabezado(archivo)
3 #eliminando la columna MME la cual seria Y
4 del(columns[0])
5
6 scaled_df = pandas.DataFrame(scaled_df, columns=columns)
7 np.seterr(divide='ignore', invalid='ignore')
8 |
9 X=np.array(X,'float')
10 Y=np.array(Y,'float')
```

	edad	Romgitano	Raizal	Palenquero	Negro	Otro	EscolarNinguna	\
0	0.34375	0.0	0.0	0.0	0.0	1.0	0.0	
1	0.09375	0.0	0.0	0.0	0.0	1.0	0.0	
2	0.09375	0.0	0.0	0.0	0.0	1.0	0.0	
3	0.09375	0.0	0.0	0.0	0.0	1.0	0.0	
4	0.12500	0.0	0.0	0.0	0.0	1.0	0.0	

	Primaria	Secundaria	Tecnica	...	I11-I15	J00-J06	J40-J47	O10-O16	\
0	0.0	1.0	0.0	...	0.0	0.0	0.0	0.0	
1	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
2	0.0	1.0	0.0	...	0.0	0.0	0.0	0.0	
3	0.0	1.0	0.0	...	0.0	0.0	0.0	0.0	
4	0.0	1.0	0.0	...	0.0	0.0	0.0	0.0	

	O20-O29	O30-O48	O95-O99	P05-P08	Q50-Q56	Z00-Z13
0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0
2	1.0	1.0	0.0	0.0	0.0	0.0

Figura 4.10 Conversión de tipo de datos

Se puede notar total resumen de los datos de las columnas de una manera difícil de analizar a primera vista, por eso esos datos también se muestran de manera gráfica de tal manera que esté representada en los valores agrupados, en este caso cabe resaltar que la imagen y las variables son muchas y no permite ser visualidad de manera correcta en su totalidad, pero es posible examinar una por una si es necesario ser observa un vistazo de la imagen:

Tipos de etiquetas de "Y" y grafica de características



Figura 4.11 Gráfica valores de características

Se pueden obtener muchos datos importantes que nos ayudarían a mejorar los modelos los cual fueron previamente entrenados, es posible también realizar otras gráficas útiles para el análisis de los datos.

En el ajuste y exploración de la data se incluyen muchos gráficos que ayudan a personas como científicos de información a tomar decisiones para tener un set de datos limpio y más preciso. En la gráfica siguiente se muestran los comportamientos que tienen las variables respeto a su dispersión lineal.

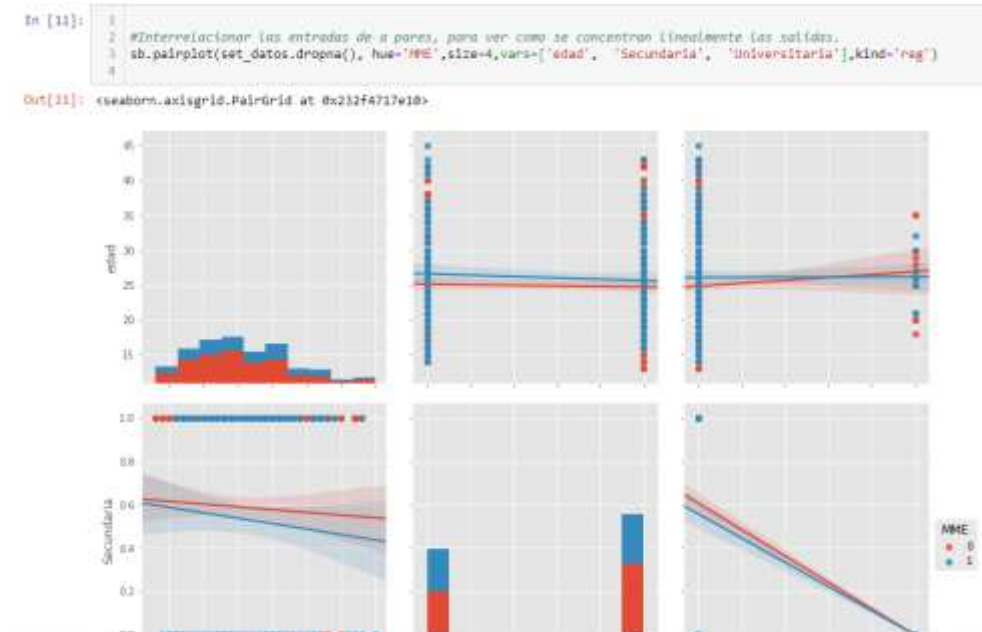


Figura 4.12 Graficas comportamiento características.

4.3.4 Selección de variables.

La técnica de selección de variables más eficiente y precisa para el modelo se realizó con Pipeline de la librería de **Sklearn** y como resultado se obtuvo de 50 variables una selección total de 10 variables el cual son las siguientes:

- MME
- EscolarNinguna
- Multiplicidad
- Sepsis
- TORCHS
- J00-J06
- J40-J47
- O10-O16
- P05-P08
- Q50-Q56

4.3.5 División del set de datos.

En el set de datos se debe tener en cuenta que los modelos más finos se ayudan entrenando y validando, de esta manera las divisiones para estos algoritmos fueron un

80% para entrenar y un 20% para validar. Se implementó una librería de Python que sirve de ayuda para hacer una división aleatoria inteligente y de esta manera compensar equivalencia en peso de clases y en cuanto etiquetas y características.

```
In [189]: 1 #División de datos 80% Entrenamiento 20% Validación
2 X_pipeline = modelo_pipeline.named_steps['anova'].transform(X)
3 validation_size = 0.20
4 seed = 7
5 X_train, X_validation, Y_train, Y_validation = model_selection.train_test_split(X_pipeline, Y, test_size=validation_size, r
+

(525, 38)
[[21, 0. 1. ... 0. 0. 1.]
 [29, 1. 1. ... 0. 0. 0.]
 [17, 0. 0. ... 0. 0. 0.]
 ...
 [23, 0. 0. ... 0. 0. 1.]
 [35, 1. 1. ... 0. 1. 0.]
 [35, 0. 1. ... 1. 0. 0.]] [0. 1. 0. 0. 0. 0. 1. 1. 0. 0. 1. 1. 0. 0. 0. 1. 0. 0. 0. 1. 0. 0. 1. 0. 1. 0. 0. 1. 0.
 0. 0. 0. 0. 0. 1. 0. 0. 1. 0. 1. 0. 0. 1. 0. 1. 0. 1. 0. 1. 0. 1. 0. 1. 0. 1. 1. 0. 1. 1.
 0. 0. 0. 1. 0. 1. 0. 1. 0. 1. 0. 1. 0. 1. 0. 1. 0. 0. 0. 0. 1. 0. 0. 0. 0. 1. 0. 0. 0. 1. 0.
 0. 1. 0. 0. 1. 0. 1. 0. 1. 0. 0. 1. 0. 1. 1. 1. 1. 1. 0. 0. 1. 1. 1. 0. 0. 1.
 0. 0. 0. 0. 1. 0. 0. 1. 0. 0. 1. 0. 1. 1. 1. 1. 1. 0. 0. 1. 1. 0. 0.
 0. 0. 1. 0. 1. 0. 0. 1. 0. 1. 0. 1. 0. 0. 0. 0. 1. 1. 1. 1. 0. 1. 1. 0. 0. 0. 0.
 0. 0. 1. 0. 1. 0. 0. 1. 0. 1. 0. 1. 0. 1. 1. 1. 1. 1. 1. 0. 0. 1. 0. 0. 0. 0.
 1. 0. 0. 1. 1. 0. 0. 0. 1. 0. 1. 0. 1. 0. 1. 1. 1. 1. 1. 1. 0. 0. 1. 0. 0.
 1. 0. 0. 0. 0. 1. 0. 0. 0. 1. 0. 0. 0. 0. 1. 1. 1. 1. 0. 0. 0. 1. 1. 1. 0. 0. 0.
 0. 0. 0. 1. 0. 0. 1. 0. 1. 0. 0. 1. 0. 1. 0. 0. 1. 1. 1. 0. 0. 0. 1.
 0. 0. 1. 0. 0. 0. 0. 0. 0. 1. 0. 1. 0. 0. 1. 0. 0. 1. 0. 1. 0. 1. 1. 1. 0. 1. 1.
 0. 1. 0. 0. 0. 1. 0. 0. 1. 0. 0. 1. 1. 1. 0. 0. 1. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 1. 0. 0. 0. 1. 1. 1. 1. 0. 0. 0. 1. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 1. 0. 0. 1. 0. 1. 0. 1. 0. 0. 0. 1. 1. 1. 0. 0. 0. 1. 0.]
```

Figura 4.13 División del set de datos

En el set de datos las características ya se encuentran normalizadas con valores flotantes e igual que las etiquetas, se conoce la dimensión total de set de datos la cual es 525 filas siendo el 80% de las 657 filas de la original en características y etiquetas, también para el set de datos de validaciones quedo con el 20% de la data general y eso sería todo en la preparación de los datos, hasta este punto ya estaría listo para aplicar los modelos de predicción.

4.3.6 Modelo de regresión logística.

Este modelo opta por ser considerado uno de los más utilizados para el aprendizaje supervisado para clasificación no lineal, la librería **sklearn** implementa muy brevemente los algoritmos, ayudando a realizar grandes soluciones de manera rápida.

Los modelos fueron probados varias veces hasta que se determine los ajustes para asegurar que las validaciones de los algoritmos fueran precisas, al final los resultados sean bastantes óptimos para ser considerados.


```
In [131]: 1 matriz = confusion_matrix(Y_validation,prediccion)
2 print(matriz)
3
4 #verdaderos positivos
5 verdaderos_positivos = matriz[0,0]
6 #verdaderos negativos
7 verdaderos_negativos = matriz[1,1]
8 #falsos positivos
9 falsos_positivos = matriz[0,1]
10 #falsos negativos
11 falsos_negativos = matriz[1,0]

[[78  2]
 [41 11]]
verdaderos_positivos 78
verdaderos_negativos 11
falsos_positivos 2
falsos_negativos 41
```

Figura 4.16 Matriz de confusión

La matriz se obtuvo de los modelos previamente entrenados con sus respectivos conjuntos de datos de entrenamiento y validación, los resultados obtenidos son:

```
In [132]: 1 sensibilidad = (verdaderos_positivos / (verdaderos_positivos + falsos_negativos) * 100) / 1
2 precision = (verdaderos_positivos / (verdaderos_positivos + falsos_positivos) * 100) / 1
3 especificidad = (verdaderos_negativos / (falsos_positivos + verdaderos_negativos) * 100) / 1

Presicion: 97.5
Sensibilidad: 65.54621848739495
Especificidad: 84.61538461538461
```

Figura 4.17 Medición del modelo regresión logística

De los resultado se observa que se tiene alta precisión, sensibilidad fue el dato menos preciso y especificidad no estuvo mal. Para este modelo se realiza la gráfica de Área de aprendizaje bajo la curva (ROC) para obtener una visión del comportamiento y poder apreciar su precisión.

```
In [136]: 1 false_positive_rate, true_positive_rate, thresholds = roc_curve(Y_validation,prediccion)
2 curva = auc(false_positive_rate,true_positive_rate)
3
4 plt.title('Receiver Operating Characteristic')
5 plt.plot(false_positive_rate, true_positive_rate, 'b',
6 label='AUC = %0.2f'% curva)
7 plt.legend(loc='lower right')
8 plt.plot([0,1],[0,1], 'r--')
9 plt.xlim([-0.1,1.2])
10 plt.ylim([-0.1,1.2])
11 plt.ylabel('True Positive Rate')
12 plt.xlabel('False Positive Rate')
13 plt.show()
```

Figura 4.18 curva ROC (modelo regresión logística)

Con la librería de **matplotlib** se muestra la gráfica de ROC del modelo de regresión logística en la siguiente gráfica:

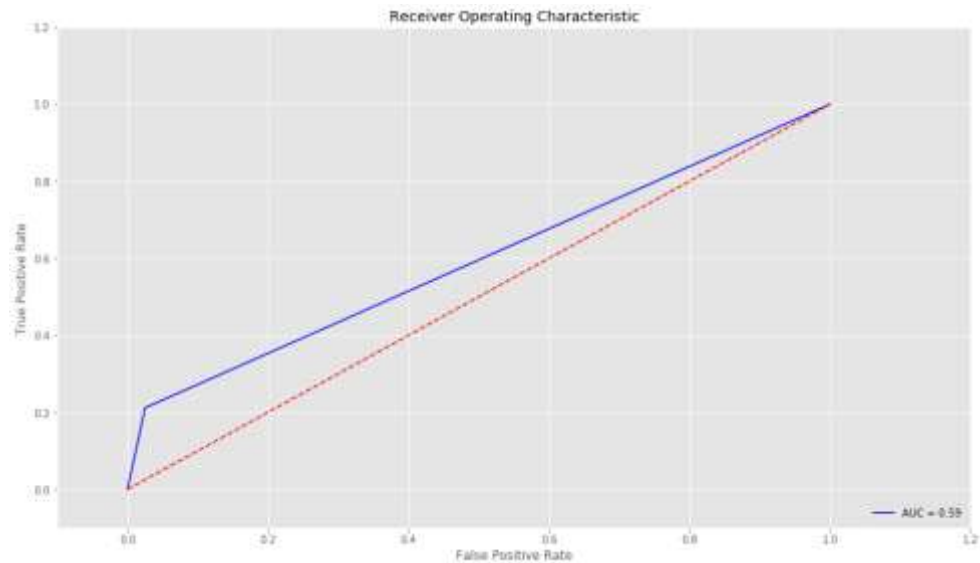


Figura 4.19 Gráfica curva ROC (modelo regresión logística)

4.3.7 Modelo de Máquinas de Soporte Vectorial.

Para el mejor de los casos algunos modelos supervisados son complicados para tratar con clasificación, por eso existen diferentes algoritmos los cuales ayudan ajustando unos más que otros, es por eso que se busca comprar cuál de los modelos es más adecuado para las predicciones con casos de hipertensión.

```
In [52]: 1 modelo_svc = SVC(decision_function_shape='ovo') #MODELO
         2 modelo_svc.fit(X_train,Y_train) #ENTRENAMIENTO

Out[52]: SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
           decision_function_shape='ovo', degree=3, gamma='auto', kernel='rbf',
           max_iter=-1, probability=False, random_state=None, shrinking=True,
           tol=0.001, verbose=False)
```

Figura 4.20 Modelo Máquinas de Soporte Vectorial

Se crea un modelo con la data de entrenamiento para poder validar las predicciones, pero primero se evalúa el modelo para saber su comportamiento de efectividad.

```
In [59]: 1 print('Evaluacion validacion:',modelo_svc.score(X_validation,Y_validation))
          2 print('Modelo: ',modelo_svc.score(X_validation,Y_validation)*100,'%')

Evaluacion validacion: 0.6212121212121212
Modelo: 62.121212121212125 %
```

Figura 4.21 Evaluación del modelo Máquinas de Soporte Vectorial

Se observa y se puede notar que tan efectivo está el modelo, pero aun así se puede hacer otras validaciones que también ayudarían a la toma de decisiones de acuerdo a los resultados de los algoritmos o predicciones.

La efectividad del modelo no es lo mismo que la precisión, normalmente se busca que los modelos estén generalizados para que puedan tener un excelente comportamiento en la predicción, en términos simples hay que buscar precisiones más altas cercanas al 100% de certeza, pero sin hacer sobre ajustes indebidos para evitar malos resultados cuando esté comparando datos reales en producción. En la siguiente graficase encuentra una clasificación para ver qué datos de la predicción y analizar sus salidas.

Clasificacion o Prediccion

```
In [62]: 1 prediccion_svc = modelo_svc.predict(X_validation)
          2 print(prediccion_svc)

[1. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0.
 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

Figura 4.22 Predicción del modelo SVC

A simple vista no se puede notar la certeza del modelo, pero puede compararse con validaciones de precisión y demás evaluadores de modelos y características, como también se puede obtener una matriz de confusión para hacer una medición.

```

In [49]: 1 matriz = confusion_matrix(Y_validation, prediccion_svc)
          2 print(matriz)
          3 #verdaderos positivos
          4 verdaderos_positivos = matriz[0,0]
          5 #verdaderos negativos
          6 verdaderos_negativos = matriz[1,1]
          7 #falsos positivos
          8 falsos_positivos = matriz[0,1]
          9 #falsos negativos
          10 falsos_negativos = matriz[1,0]

[[76  4]
 [46  6]]
verdaderos_positivos 76
verdaderos_negativos 6
falsos_positivos 4
falsos_negativos 46

```

Figura 4.23 Matriz de Confusión Máquinas de Soporte Vectorial

con esta información se aplican las fórmulas de precisión, sensibilidad y especificidad para evaluar frente a otros ajustes del modelo u otros modelos y poder comparar resultados.

Aprovechando que **sklearn** tiene la oportunidad de aplicar algoritmos que sacan los valores aplicamos las fórmulas las cuales miden la calidad de los resultados de los modelos que se obtuvieron.

```

In [63]: 1 sensibilidad_svc = (verdaderos_positivos / (verdaderos_positivos + falsos_negativos) * 100) / 1
          2 precision_svc = (verdaderos_positivos / (verdaderos_positivos + falsos_positivos) * 100) / 1
          3 especificidad_svc = (verdaderos_negativos / (falsos_positivos + verdaderos_negativos) * 100) / 1

Presicion: 95.0
Sensibilidad: 62.295081967213115
Especificidad: 60.0

```

Figura 4.24 Medición del modelo Máquinas de soporte vectorial

De estos resultados se pueden tomar muchas decisiones cómo ajustar más los modelos, alimentar más la información las características y el todo el set de datos. También se puede realizar gráficas que permiten visualizar los resultados y así ver qué otras

diferencias que visualmente se pueden deducir e interpretar mucho mejor para las decisiones, una de la gráfica que ayuda a ver el comportamiento de estas notablemente es la de ROC (MSV) en cuál es la siguiente:

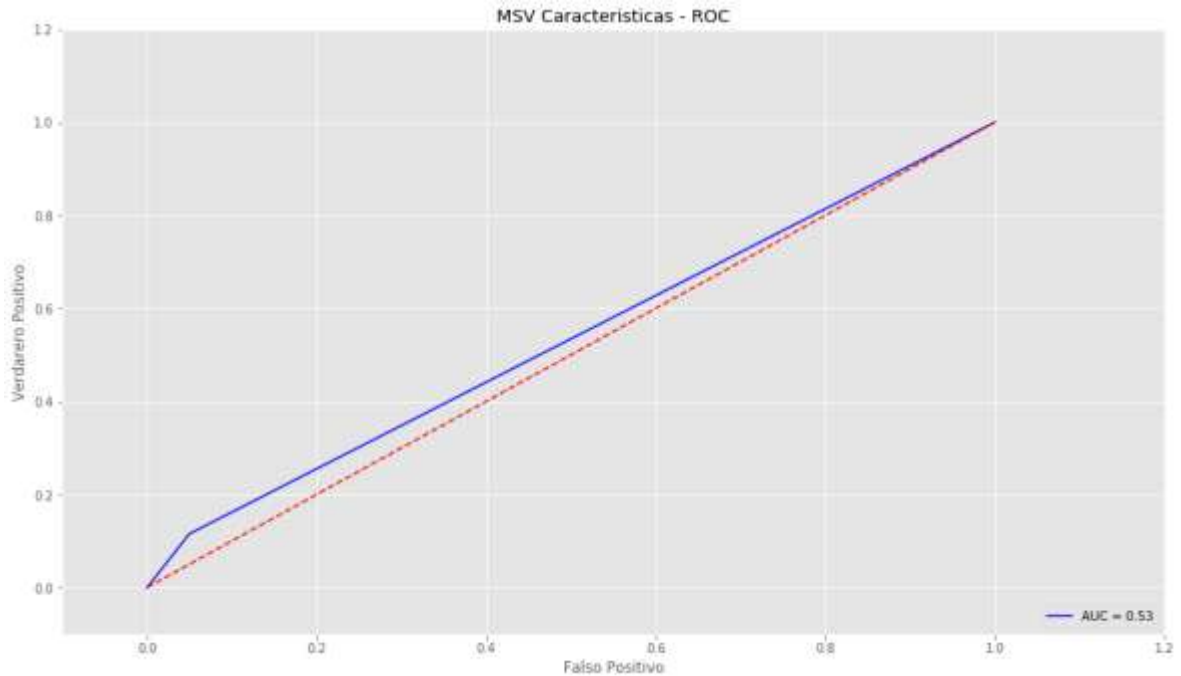


Figura 4.25 Gráfica curva de ROC

Se puede ser analistas de información si dedica a tratar los datos de para obtener mejores predicciones, es decir ver comportamientos de las variables seleccionadas y estudiar sus patrones de clases. Las predicciones son mejores cuando manejan mejores clasificadores o atributos y más características etiquetadas, luego si todo estos entrenamientos y validaciones podemos llevar a pruebas reales con características que se quieran evaluar con los modelos.

5. COMPARACIÓN DE MODELOS

5.1 RESULTADOS DE LAS VALIDACIONES DE LOS MODELOS

5.1.1 Características de Máquinas de soporte vectorial y Regresión logística

Los modelos de máquinas de soporte vectorial se destacan por que se utiliza para clasificar y hacer regresiones eficientes donde las características son mucho más en cuanto a dimensión haciendo estas más versátil y eficiente en predicciones. Pero con los resultados que se obtuvieron de las pruebas se puede decir que no se puede evaluar un modelo simplemente porque a otros estudios analíticos les ha ido muy bien y en realidad es que para poder decidir los mejores modelos se deben probar la familia de modelos y algoritmos y comparar los resultados. Es necesario probar muchas veces un solo modelo con diferentes tablas de información para saber los comportamientos de clasificación y validar que sus resultados sean los más certeros a la realidad.

Para la comparación se implementaron tres métodos de selección de variables y se aplica igualmente dos técnicas de normalización diferentes para así tener diferentes subconjuntos de los datos hasta 4 subconjuntos para comparar con cada uno de los modelos y sus resultados.

Primero se evaluarán las predicciones individualmente cada modelo para ver con cuáles técnicas se comporta mejor y luego se hará una comparación entre los modelos de regresión logística y máquinas de soporte vectorial.

Se validaron tres técnicas de selección el cual se evaluarán cada una para decidir cuál fue mejor, la primera técnica es una selección basada en el árbol el cual hace una estimación para ajustarse a decisiones aleatorias utilizando el promedio para mejorar la precisión predictiva y los sobre ajustes controlados.

ExtraTreesClassifier el selector de variables utilizado de sklearn minimiza de un total de 49 características a un subconjunto de 17 y su medición de precisión es de un 98,32% considerado alto para las demás selecciones.

```

1 X_sinnormalizar = obtener_x()
2 Y_sinnormalizar = set_datos['MME']
3 print(X_sinnormalizar.shape)
4 # Tree-based feature selection
5 seleccion_arbol = ExtraTreesClassifier()
6 entrenamiento_seleccion_arbol = seleccion_arbol.fit(X_sinnormalizar, Y_sinnormalizar)
7 entrenamiento_seleccion_arbol.feature_importances_
8
9 modelo_tree = SelectFromModel(entrenamiento_seleccion_arbol, prefit=True)
10 X_tree = modelo_tree.transform(X_sinnormalizar)
11 print(X_tree.shape)
12 print(seleccion_arbol.score(X_sinnormalizar, Y_sinnormalizar))

```

(657, 49)
(657, 17)
0.9832572298325722

Figura 5.1 Selección de variables ExtraTreesClassifier

La segunda técnica de selección de variables se implementó un evaluador lineal para seleccionar las características más importantes para el modelo que busca afinar, este obtuvo un total de 19 características seleccionadas de 49 del set de datos original.

```

1 lsvc = LinearSVC(C=0.01, penalty="l2", dual=False)
2 entrenamiento_lsvc = lsvc.fit(X_sinnormalizar, Y_sinnormalizar)
3 modelo_lineal = SelectFromModel(entrenamiento_lsvc, prefit=True)
4 X_LinearSVC = modelo_lineal.transform(X_sinnormalizar)
5 print(X_sinnormalizar.shape)
6 print(X_LinearSVC.shape)
7 print(lsvc.score(X_sinnormalizar, Y_sinnormalizar))

```

(657, 49)
(657, 19)
0.6788432267884322

Figura 5.2 Selección de variables LinearSVC

En la tercera técnica de selección se realiza una selección de modelo afinado para seleccionar las características evaluadas por los estimadores y transformaciones secuenciales de los modelos Pipeline siendo está más ajustado para los modelos.

```

1 #Creamos el modelo Pipeline - Normalización
2 modelo_pipeline = Pipeline([('anova', SelectKBest(f_regression, k=5)), ('svc', svm.SVC(kernel='linear'))])#resto parámetros pa
3 #Entrenamos nuestro modelo
4 modelo_pipeline.set_params(anova__k=10, svc__C=0.1).fit(X_sinnormalizar, Y_sinnormalizar)
5 #Clasificamos
6 prediction = modelo_pipeline.predict(X_sinnormalizar)
7
8 X_pipeline = modelo_pipeline.named_steps['anova'].transform(X_sinnormalizar)
9 print(X_sinnormalizar.shape)
10 print(X_pipeline.shape)
11 print(modelo_pipeline.score(X_sinnormalizar, Y_sinnormalizar))

```

(657, 49)
(657, 10)
0.680365296803653

Figura 5.3 Selección de variables Pipeline

Se puede evidenciar que de las técnicas de selección de variables la de árbol es la más ajustada al modelo implementados. Pero en la evaluación de los modelos tendremos más resultados que ayudan a el ajuste o generación de los modelos,

Los resultados de las técnicas de selección tomadas para la preparación de los modelos fueron sujeto a las características de las técnicas que se implementaron de sklearn el cual tiene la siguiente muestra:

Técnicas de selección	
Nombre	Características seleccionadas
Árbol	17
Lineal	19
Pipeline	10
Ninguna	49

Tabla 4 Resultados de técnicas de selección de variables

Con el conjunto de datos para validación de los modelos se hacen los ajustes necesarios para las mejoras, pero primero antes se normalizaron los datos con la librería de sklearn para ayudar en afinamiento o precisión de los modelos.

En la realización del desarrollo se implementó una normalización el cual la van hacer tenidas en cuenta, para comparar los diferentes resultados obtenidos, estas a primera son validadas por los evaluadores de cada uno de los modelos que fueron comparados, los resultados son:

Técnica	Precisión predictiva StandardScaler	
Nombre	SVC	Regresión Logística
Árbol	64,39	68,18
Lineal	68,18	69,69
Pipeline	68,18	66,66
Ninguna	66,66	65,15

Tabla 5 Precisión de los subconjuntos StandardScalers

Se implementó una normalización a todos los subconjuntos de datos y su evaluación varía para cada subconjunto, el cual algunos son mejores y equilibrados que otros. En esta primera tabla se muestra el resultado de la precisión predictiva de cada uno de los modelos que se comparan y sus resultados son más altos que otras normalizaciones.

Técnica	Precisión predictiva Normalizer	
Nombre	SVC	Regresión Logística
Árbol	60,6	60,6
Lineal	60,6	68,93
Pipeline	60,6	60,6
Ninguna	60,6	60,6

Tabla 6 Precisión de los subconjuntos Normalizer

Los datos normalizados suelen ayudar a la mejora de los modelos sin embargo como se ve en la tabla anterior respecto a la normalización anterior, se ha perdido calidad de precisión en los datos el cual a veces es bueno saber escoger las técnicas para mejorar y afinar todos los ajustes.

Técnica	Precisión predictiva MaxMinScaler	
Nombre	SVC	Regresión Logística
Árbol	65,90	69,69
Lineal	65,15	68,18
Pipeline	66,66	65,42
Ninguna	62,12	67,42

Tabla 7 Precisión del subconjunto MaxMinScaler

Probar los modelos con diferentes datos es importante, porque de esta manera se observa el comportamiento que tienen los modelos, pero sin embargo normalizando los datos para asegurar de colocar las características en el mismo formato para mejorar la precisión de predicción de cada uno, sin embargo, también se realizó una evaluación de los datos sin normalizar para comparar resultados.

Técnica	Precisión predictiva	
Nombre	SVC	Regresión Logística
Árbol	66,66	68,93
Lineal	65,90	69,69
Pipeline	62,12	67,42
Ninguna	65,66	66,66

Tabla 8 Precisión de los subconjuntos

Los diferentes subconjuntos de datos fueron analizados bajo las características de los modelos que se van a comparar, pero antes los datos fueron tratados con diferentes técnicas para ajustar mejor la precisión predictiva. Primero se aplicaron tres técnicas de selección de variable a la data original lo cual reduce por los menos 4 conjuntos de datos, luego a esos conjuntos características seleccionadas se aplica tres técnicas de normalización lo cual deja la posibilidad de comparar resultados valiosos para los ajustes del modelo,

Para los datos no normalizados se puede decir que las mejores características las tiene el subconjunto que se basó en árbol de decisión para escoger las variables en el modelo de máquinas de soporte vectorial.

Para la regresión logística los subconjuntos de datos de características basados en árbol no tienen el mismo comportamiento que con todas las características y el resto tampoco, como se muestra en la siguiente gráfica.

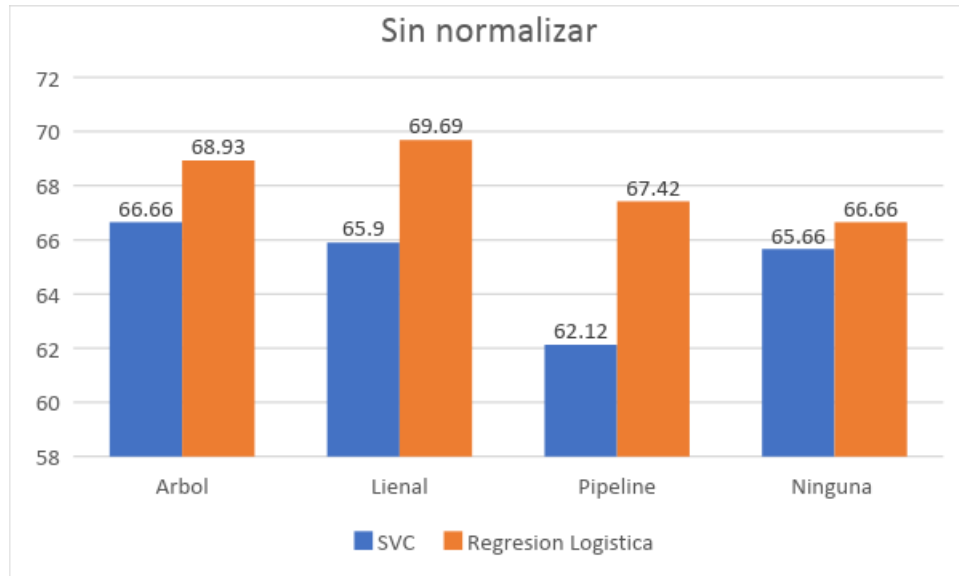


Figura 5.4 Comparación de modelos en subconjuntos

Los resultados como estos dan más argumentos que demuestran que no todas las técnicas que se aplican a los datos ayudan a la mejora de nuestros modelos por ende hay que probar con varios hasta encontrar el que más se adapta a nuestro ajuste de generalización.

Pero este apenas es un análisis de datos sin normalizar ahora visualizamos los resultados obtenidos con datos normalizados.

En esta normalización claramente la selección lineal es más alta para el modelo de máquinas de soporte vectorial, pero no dice definitivamente que es el que más se ajusta por eso mirando el de regresión logística que es el siguiente, se puede comparar cuál de los dos, esta normalización le sienta mejor.

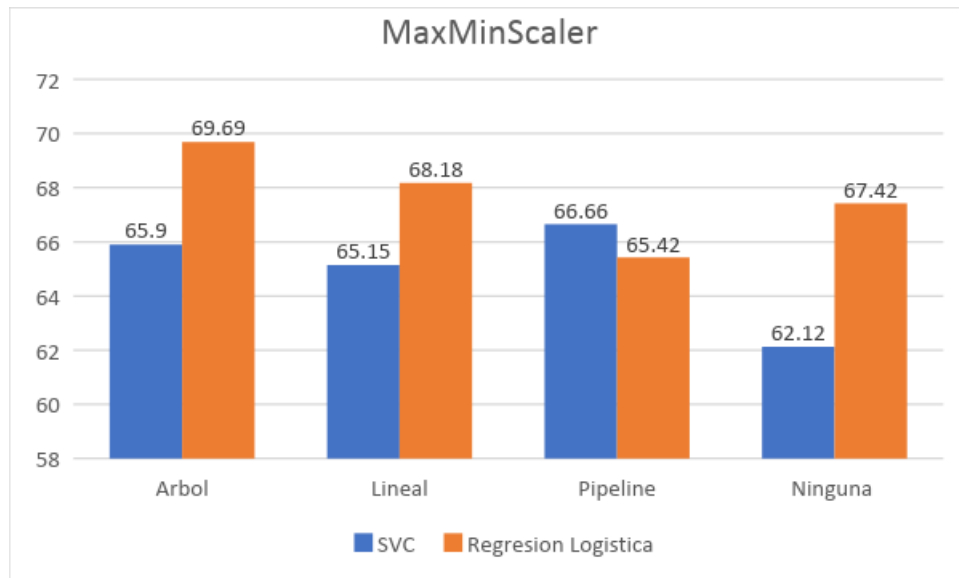


Figura 5.5 Comparación de modelos en subconjuntos MaxMinScaler

De esta manera se puede ver resultados de la normalización de MinMaxScaler, que en el modelo de regresión logística tiene mejor comportamiento con las características de las variables basadas en árbol diferente al de máquinas de soporte vectorial.

Se puede ver diferentes tipos de normalizaciones y los resultados pueden ser mejor e incluso iguales para diferentes subconjuntos, a continuación, se muestra la normalización de StandardScaler de sklearn, el cual también ayuda a darle mejor formato a los datos de características y ajustes como ya se han identificado.

Primero se evalúa el modelo de máquinas de soporte vectorial con la normalización en todos los subconjuntos de características implementadas y luego de regresión logística.

Para el modelo de máquinas de soporte vectorial la mejor opción de los subconjuntos de datos es el que está basada en árbol para esta normalización y para regresión logística es evidente en la siguiente comparación que tiene mejores resultados cuando esta contiene todas las características o no aplica un método de selección.

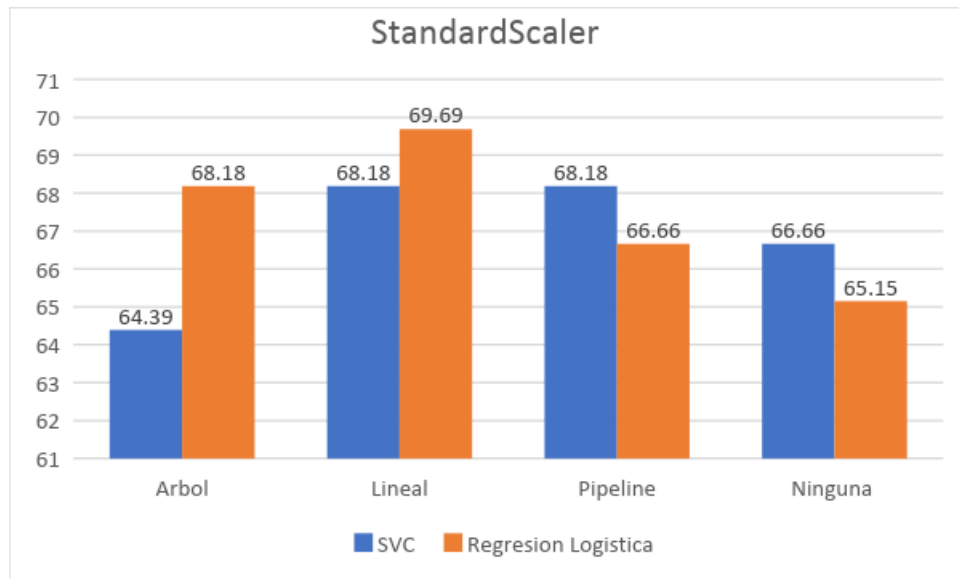


Figura 5.6 Comparación de modelos en subconjuntos normalizados con StandardScaler

Para algunos modelos es mejor trabajar con más características y que los métodos de selección de variables no son más que ayudas para cuando son necesarios hacer que los modelos sean más rápidos entrenando y aprendiendo, pero sin echar de menos la precisión y validez.

Por último, se comparan las características normalizadas por Normalizer de sklearn el cual no está demás verificar que esta normalización sea la que posiblemente menos se ajuste o, al contrario, pero para eso se hizo la comparación de los siguientes resultados:

Se puede decir que al normalizar los datos con esta técnica es igual para todos los subconjuntos y que no se ajusta a las características el cual no deja claro su mejora para el modelo. En cuanto al otro modelo la técnica también no es la mejor para tener en cuenta dado que sus resultados son similares, únicamente cambiando en el subconjunto basado en variable lineal como se muestra en la siguiente gráfica.

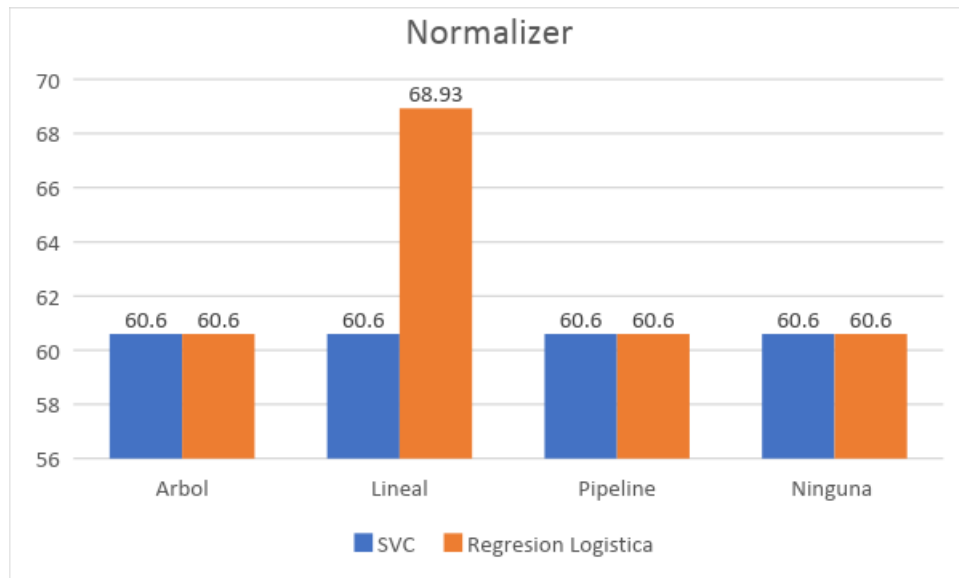


Figura 5.7 Comparación de modelos en subconjuntos normalizados con Normalizer

Ahora para tomar los subconjuntos de datos mejor ajustados para los modelos se hace una comparación individual de todos los resultados por técnicas o subconjuntos.

Se tiene que para el modelo de Máquinas de soporte vectorial la técnica que más se ajusta es la normalización **StandardScaler** principalmente con las variables seleccionadas que se basan en árbol y se podría decir otros análisis a partir de comparaciones importantes. La siguiente tabla muestra el conjunto de resultado de evaluaciones realizadas al modelo de máquinas de soporte vectorial con los diferentes subconjuntos, se representan en técnicas y subconjuntos y se evalúa el modelo con cada técnica y normalización, ayudando a tomar decisiones en los sets de datos a generalizar.

Máquina de soporte vectorial				
Técnica	Árbol	Linear	pipeline	Ninguna
StandardScaler	78,78	75,75	69,69	75,75
Normalizer	60,6	60,6	60,6	60,6
MinMaxScaler	66,66	67,42	66,66	62,12
Ninguna	73,48	67,42	66,66	66,66

Tabla 9 Resultados de precisión de todos los subconjuntos en modelo SVC

Para este modelo su inclinación está llevado a utilizar una técnica para subir la precisión predictiva, ajustando cada subconjunto con una normalización. Una comparación que hace más visible las diferencias de estas evaluaciones se muestra así:

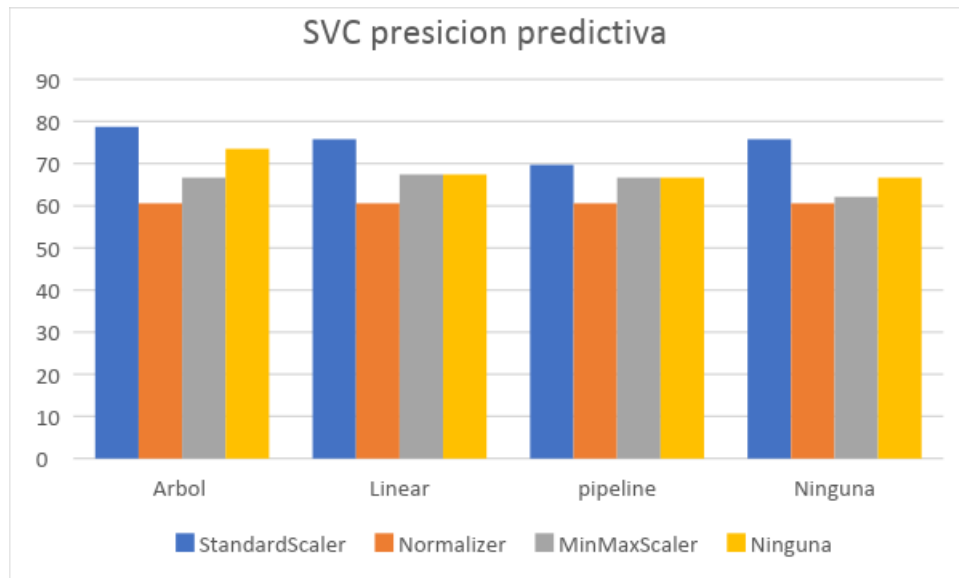


Figura 5.8 Comparación de precisiones normalizadas con cada subconjunto

Viendo los comportamientos de los subconjuntos resalta que el subconjunto de árbol tiene mejor comportamiento con las características seleccionadas. No es posible decir lo mismo para el otro modelo el cual implementa un algoritmo diferente.

En la comparación del modelo de regresión logística con los subconjuntos de datos y las diferentes normalizaciones se identifica poca variedad en la efectividad del modelo, pero sí una buena conclusión en cuanto a optimización o rendimiento. Los resultados de las evaluaciones del modelo se inclinan a que solo debe tener una técnica implementada y que esta sería como la referencia para los demás datos o características que se evalúan. Para regresión logística los resultados fueron los siguientes:

Regresión Logística				
Técnica	Árbol	Linear	pipeline	Ninguna
StandardScaler	70,45	69,69	68,93	71,21
Normalizer	60,6	68,93	60,6	60,6
MinMaxScaler	71,21	68,93	68,93	70,45
Ninguna	71,21	69,69	68,93	71,21

Tabla 10 Resultados de precisión en todos los subconjuntos normalizados

Se puede notar que para todas las características la precisión de predicción es más alta que aplicando otras técnicas de selección, es decir con este modelo trabajar con todas las características es igual que aplicar una normalización de MinMaxScaler y aun así también los resultados son iguales en cuanto a predicción.

No hay mucho que detallar de los análisis realizados, está claro que para este modelo solo hay dos alternativas para manejar los datos de las características y en la siguiente gráfica se obtiene mejor visibilidad de las comparaciones de los subconjuntos y normalizaciones.

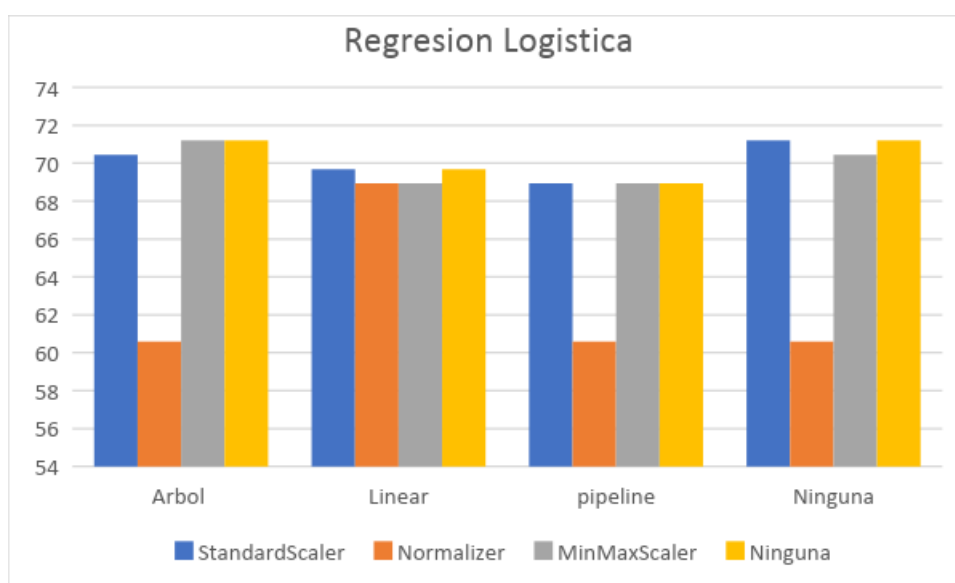


Figura 5.9 Comparación de precisión del modelo regresión logística en todos los subconjuntos

Claramente también puede analizar que la normalización con Normalizer de sklearn es muy baja para algunos subconjuntos y que este modelo trabaja mejor con todas las características y no todas las normalizaciones.

Por último, se comparan los resultados de los modelos en cuanto precisión sensibilidad y especificidad de la cual se obtuvo cuatro resultados diferentes que se pueden elegir para aplicar a los modelos que mejor están ajustados. Para el modelo de máquinas de soporte vectorial le va muy mal en las predicciones según las medidas realizadas en este último análisis, teniendo como consecuencia más mediciones de los modelos es notable que los

datos del algoritmo o características se adaptan mejor a la regresión logística para la tabla de resultados de máquina soporte vectorial se obtiene:

SVC			
	precisión	sensibilidad	especificidad
StandardScaler	96,25	66	81,25
Normalizer	100	60,6060606061	0
MinMaxScaler	95	65,5172413793	75
Sin normalizar	95	65,5172413793103	75

Tabla 11 Resultados de precisión, sensibilidad y especificidad (SVC).

Se puede notar la torpeza de este modelo para encajar en una mejor especificidad sin embargo en precisión sube bastante, pero queda muy sobre ajustados por ende no es lo mejor para este tipo de predicciones o clasificaciones.

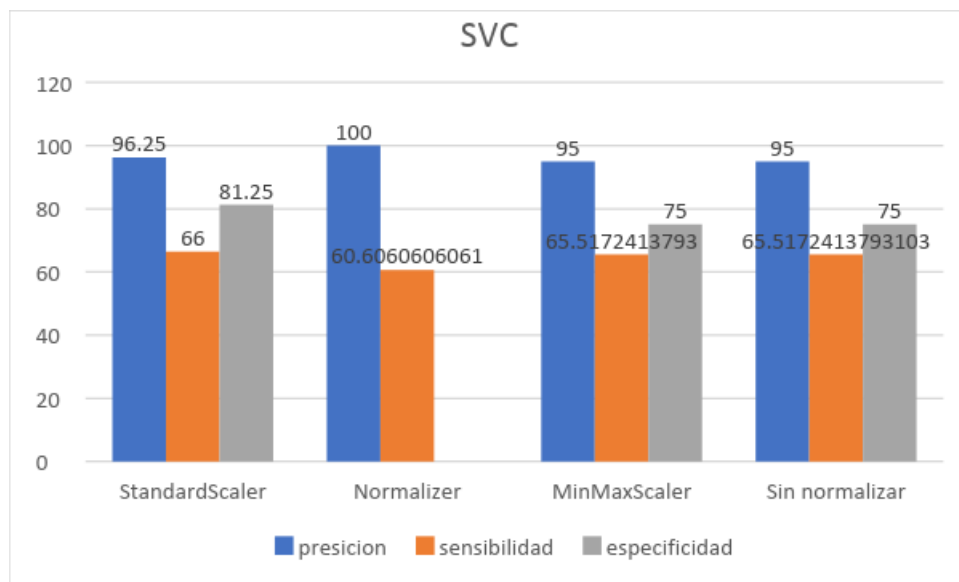


Figura 5.10 Comparación de resultados de medición con todos los subconjuntos en SVC

En la gráfica se ve mucho más claros los datos que se obtenidos de las mediciones el cual para cada subconjunto de datos los resultados varían. Por otro lado, en el modelo de regresión logística los datos se ven muy bien en cuanto las mediciones que hacemos para validar cada modelo comparado, tenemos que:

Regresión Logística

	precisión	sensibilidad	especificidad
StandardScaler	96.25	67,54385965	83,33333333
Normalizer	96.25	66,37931034	81,25
MinMaxScaler	97,5	65,54621849	84,6153846
Sin normalizar	97,5	66,66666667	86,66666667

Figura 5.11 resultados de mediciones del modelo regresión logística en todos los subconjuntos

Como se puede ver en la tabla anterior este modelo tiene mejor comportamiento para la predicción, este está más equilibrado en todas las variables de medición por ende también se porta mejor para la predicción.

En este punto se nota que la precisión es mucho más estable que el otro modelo de SVC y también se puede ver la variación con los diferentes tipos de subconjuntos tratados para la mejora de cada característica. La especificidad es bastante alta, pero esta es más alta cuando el número de características es mayor y de esta forma se compensa otras mediciones igualmente.

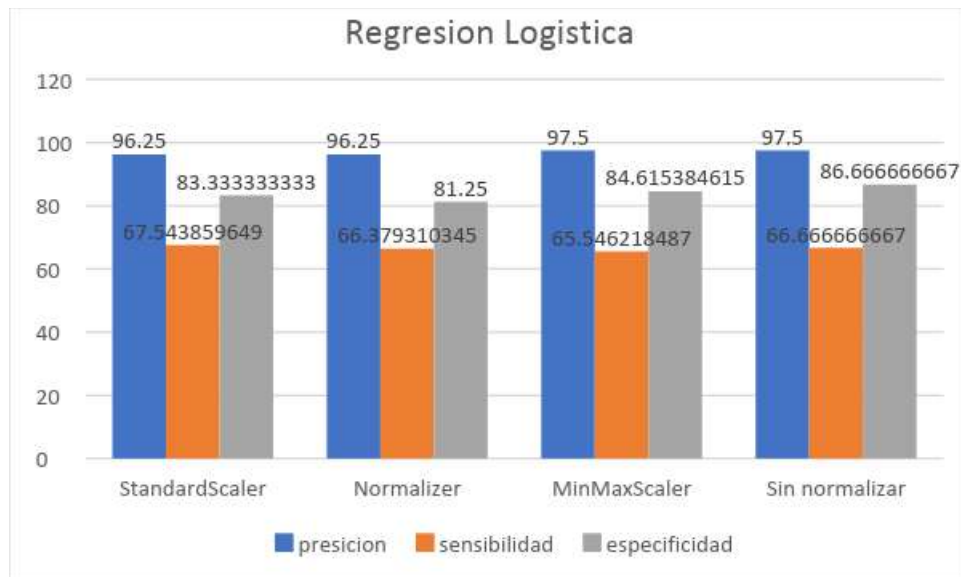


Figura 5.12 Comparación de mediciones del modelo regresión logística en todos los subconjuntos

Luego de toda la preparación ya se tiene los modelos listos para hacer clasificaciones y predicciones que pueden ayudar en la decisión de evaluaciones. En cuanto a estos modelos comparados el más correcto para la predicción es el de regresión logística, este tuvo mejor resultado de medición.

6 RECOMENDACIONES

Las recomendaciones que aportarían más para este proyecto son las de buscar más modelos y construir un conjunto de datos mucho más grande para el entrenamiento de los modelos, el cual el aprendizaje de ellos va ligado a el conocimiento masivos de casos reales presentados para contar con eso como un validador de clasificaciones. El aprendizaje automático no se adapta a las ideas de negocios de cualquier información, estas deben ser estudiadas y analizadas por expertos en datos para modelar esos patrones de características que se deben generalizar en un modelo para que aprenda automáticamente cada vez que una nueva información o un nuevo caso aparezca.

7 CONCLUSIONES

Según los resultados obtenidos a través de las pruebas realizadas entre los modelos de aprendizajes automáticos según las variables como sensibilidad precisión y especificidad se puede concluir que el algoritmo con mejor comportamiento y mejores resultados obtenidos con el set de datos utilizados es regresión logística, teniendo en cuentas las pruebas y los resultados obtenido mediante los diferentes técnicos de normalización aplicadas se afirmar que los algoritmos de regresión logística nos proporcionaron un mejor resultado óptimo y más satisfactorio para nuestro caso de estudio.

De los modelos comparados el más apto para la predicción de casos de personas con hipotensión sería el modelo de regresión logística, el cual tiene una alta precisión y especificidad, pero en sensibilidad a pesar no está alto tampoco está mal, esto le permite estar entre un modelo casi generalizado pero que aún le faltaría un conjunto de datos muchos más amplio, también hay que tener en cuenta que los datos también influyen en el entrenamiento de las etiquetas, los datos por otra parte deberían ser balanceados en cuanto a etiquetas del aprendizaje supervisado. Para el modelo de máquina de soporte vectorial se salió del ajuste y pasó a tener una precisión de 100% el cual hace el modelo sobre ajustado y no apto para este caso de implementación en la salud.

Para este trabajo que involucra una serie de temas importantes para el mundo de salud y la informática resaltamos lo bueno que es trabajar con datos e información clasificadora para realizar soluciones de problemas que se han presentado alguna vez y que han sido analizadas por algoritmos que clasifican mejor los datos. Hoy en día existen muchas herramientas que ayudan a la predicción de acontecimientos que se basan hechos y para este caso la conclusión está en que hay probar todas las herramientas para saber cuál se comporta y lo hace mejor.

Los algoritmos que se implementaron para la comparación de modelos están en la librería de sklearn de Python estas vienen ya incluidas por defecto en el entorno de desarrollo de anaconda. para mejorar mucho más la predicción de los modelos se debe aumentar el número de datos de del conjunto de datos para así afinar la precisión y otros evaluadores de nuestros modelos.


Bibliografía

- [1] J. T. P. M. Roque Luis Marín Morales, Inteligencia artificial. Técnicas, métodos y aplicaciones, España: McGraw-Hill Interamericana de España S.L, 2008, p. 10022.
- [2] M. A. C. Q. O. C. P. F. E. R. M. A. L. O. Maria Isabel Alfonso Galipienso, Inteligencia artificial: modelos, técnicas y áreas de aplicación, Esáña, Madrid: Consuelo Garcia Asensio y Clara M, 2003.
- [3] Eugenia Luz Arrieta Rodriguez, Prediccion Temprana de Morbilidad Materna Extrema Usando Aprendizaje Automatico, Cartagena: Universidad Tecnologica de Bolivar Facultad de Ingenierias Cartagena, 2017.
- [4] U. A. Valbuena GJP, «Una mirada a los gastos de bolsillo en salud para Colombia,» *Banco Repúb*, p. 27–56, 2017.
- [5] G. V. Kattah AG, «The Management of Hypertension in Pregnancy,» *Chronic Kidney Dis*, vol. 3, pp. 229-39., 2013.
- [6] G. V. Kattah AG, The Management of Hypertension in Pregnancy, *Adv Chronic Kidney Dis*, 2013.
- [7] G. C. G. D. D. L. L. B. J. C. C. De Sereday MS, Prevalence of diabetes, obesity, hypertension and hyperlipidemia in the central area of Argentina, Argentina: Diabetes Metab, 2004.
- [8] E. L. A. Rodríguez, Predicció n Temprana de Morbilidad Materna Extrema Usando Aprendizaje, Cartagena, 2017.
- [9] S.-D. N. Díaz-Martínez LA, Oportunidades de investigación en preeclampsia, desde la perspectiva de prevención primaria. un artículo de reflexión a paper aimed at, *Rev Colomb Obstet Ginecol*, 2008.
- [10] K. N. C. T. L. A. N. K. Poon LCY, Maternal risk factors for hypertensive disorders in pregnancy: a multivariate approach, *J Hum Hypertens*, 2010.
- [11] E. O. Barker LR, Manejo de la hipertensión en consulta externa del Hospital Universitario del Valle, Cali, 2017.
- [12] C. Colombiana, Ley 1221, Bogota, 2008.
- [13] P. FHG, «La crisis de la salud en Colombia: un problema moral,» *Colomb Salud Libre*, vol. 1, p. 48–56, 2017.
- [14] P. N. y. S. J. Russell, Inteligencia Artificial: Un Enfoque Moderno, Prentice Hall, 1994.

- [15] R. M. M. José T. Palma Méndez, Inteligencia artificial. Técnicas, métodos y aplicaciones, latinoamerica, 2008.
- [16] V. Mathivet, Inteligencia Artificial para desarrolladores, ENI, 2015.
- [17] P. M. G. y. R. L. d. Mantaras, Inteligencia artificial, 2017.
- [18] S. V. H. A. R. Carlene M M Lawes, Global burden of blood-pressure-related disease, New Zealand , 2001.
- [19] M. Á. C. A. B. M. Muñoz OM, Concordancia entre los modelos de SCORE y Framingham y las ecuaciones AHA/ACC como evaluadores de riesgo cardiovascular, Rev Colomb Cardiol, 2017.
- [20] A. Turing, Computing machinery and intelligence, Londres: Smith's Prize, 1950.
- [21] C. J. R. J. Saldaña J, Caracterización de complicaciones cardiovasculares por hipertensión arterial en afiliados a una EPS privada en la ciudad de Neiva en el año 2011, 2017, Salut Sci Spirit.
- [22] A. P. Castaño, Aprender Inteligencia Artificial, Combinatoria, Grafos y Algoritmos en Python, Cuba, 2018.
- [23] Y. H.-X. W. Y.-M. Z. W.-W. M. W.-Y. W. Y.-Q. Zhu Y-C, Analysis of correlation factors and pregnancy outcomes of hypertensive disorders of pregnancy- a secondary analysis of a random sampling in Beijing, China, 2016.

ANEXOS

ANEXO A. Presupuesto del proyecto

 UNIVERSIDAD DEL SINÚ Elías Bechara Zainúm Seccional Cartagena	PROCESO: INVESTIGACIÓN, CIENCIA E INNOVACIÓN TÍTULO: PRESUPUESTO PROYECTO DE INVESTIGACIÓN CODIGO: R-INVE-030 VERSIÓN: 002				
Título del proyecto:					
Nombre del grupo:					
Rubro	Recursos Unisinu Cartagena		Recursos Externos		Total
	Especie	Frescos	Especie	Frescos	
Personal	\$ 1.440.000,00	\$ -	\$ 6.090.000,00	\$ -	\$ 7.530.000,00
Servicios técnicos	\$ -	\$ -	\$ -	\$ -	\$ -
Equipos de uso propio	\$ -	\$ -	\$ 3.700.000,00	\$ -	\$ 3.700.000,00
Compra de equipos	\$ -	\$ -	\$ -	\$ -	\$ -
Materiales / insumos / reactivos	\$ -	\$ -	\$ -	\$ -	\$ -
Salidas de campo	\$ -	\$ -	\$ -	\$ -	\$ -
Software	\$ -	\$ -	\$ -	\$ -	\$ -
Viajes	\$ -	\$ -	\$ -	\$ -	\$ -
Gastos de publicación	\$ -	\$ -	\$ -	\$ -	\$ -
Gastos de patentes	\$ -	\$ -	\$ -	\$ -	\$ -
Total	\$ 1.440.000,00	\$ -	\$ 9.790.000,00	\$ -	\$ 11.230.000,00
TOTAL					\$ 11.230.000,00
Caracterización de la inversión	Entidades		Total	Especie	Frescos
	Inversión unisinu		13%	13%	0%
	Inversión externa		87%	87%	0%