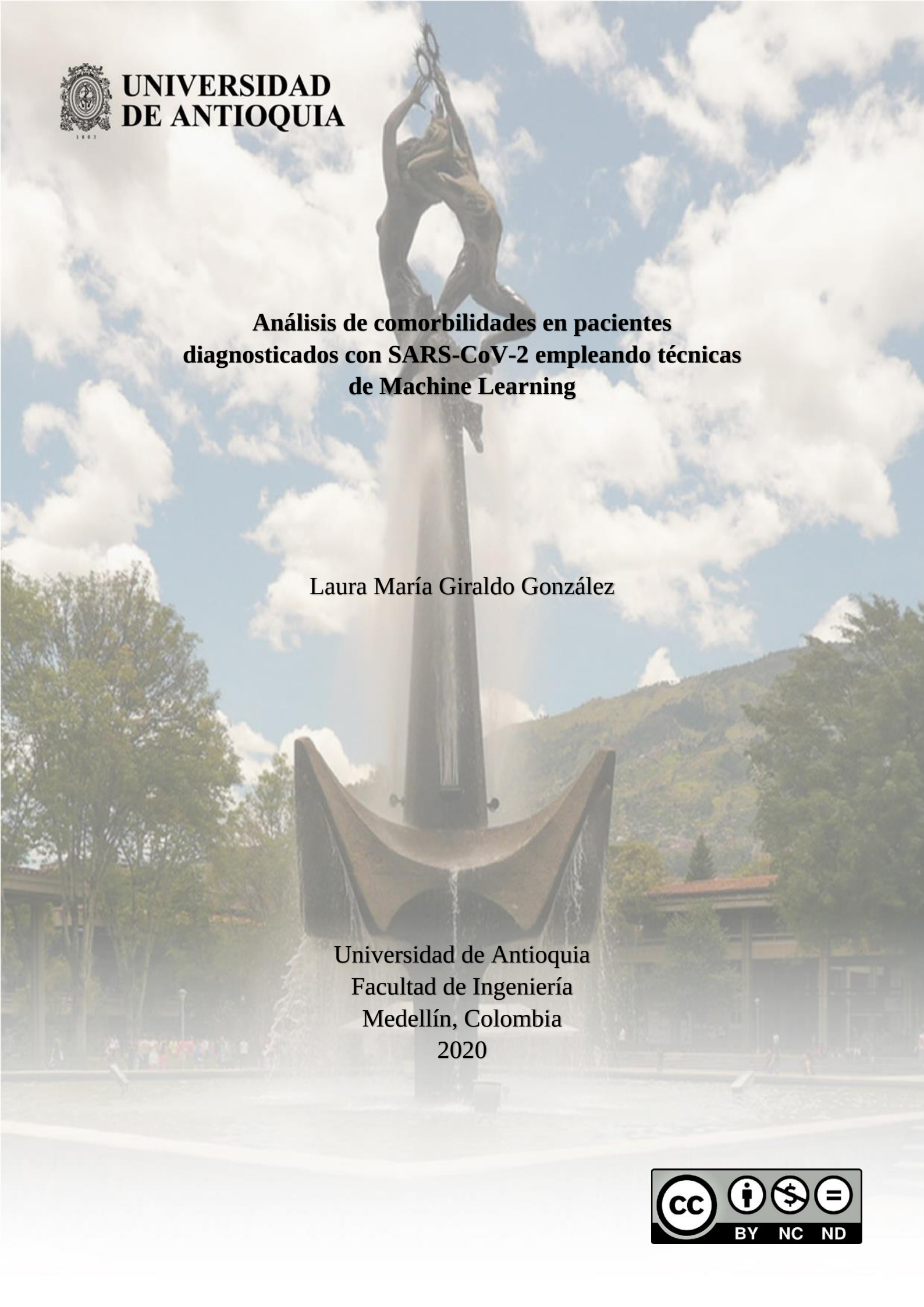

UNIVERSIDAD DE ANTIOQUIA

Análisis de comorbilidades en pacientes diagnosticados con SARS-CoV-2 empleando técnicas de Machine Learning

Laura María Giraldo González

Universidad de Antioquia
Facultad de Ingeniería
Medellín, Colombia
2020

Análisis de comorbilidades en pacientes diagnosticados con SARS-CoV-2 empleando técnicas de Machine Learning

Laura María Giraldo González

Trabajo de investigación presentado como requisito parcial para optar al título de:
Bioingeniera

Asesor
María Bernarda Salazar Sánchez, Ph.D.

Universidad de Antioquia
Facultad de Ingeniería
Medellín, Antioquia
2020

Agradecimientos

Quiero agradecerle principalmente a la profesora María Bernarda Salazar Sánchez por su constante acompañamiento durante el desarrollo del presente proyecto, el cual fue clave para enfrentar los problemas que se presentaron a lo largo de este. Así mismo, quiero expresarle mis agradecimientos a Luis Felipe Buitrago Castro, por la orientación que me brindó sobre las técnicas de Machine Learning y por su permanente disposición.

Adicionalmente, agradezco a María Mesa Rivera de la revista Semana por proporcionarme la información necesaria para el desarrollo del proyecto y finalmente, quiero agradecerle a mi familia por el apoyo incondicional durante este proceso.

Índice de contenido

Índice de figuras

Índice de tablas

1 Resumen

La Organización Mundial de la Salud ha catalogado a la enfermedad generada por el virus SARS-CoV-2 (COVID-19) como una emergencia sanitaria para la salud pública de gran impacto y relevancia a nivel internacional [1]. Aunque alrededor del 80% de las personas contagiadas se recuperan de la enfermedad sin necesidad de tratamiento hospitalario, aproximadamente una de cada cinco personas, termina presentando un cuadro grave, siendo los más afectados, las personas que padecen afecciones médicas previas como hipertensión arterial, problemas cardiacos, enfermedades pulmonares, diabetes y cáncer [2]. Por tanto, es fundamental comprender la relación que existe entre la muerte de pacientes a causa del COVID-19 y sus enfermedades de base, lo que permitirá identificar oportunamente la población de riesgo.

Por tanto, con el objetivo de comprender mejor la relación entre la enfermedad y patologías de base, en el presente proyecto se implementaron cinco modelos mediante la aplicación de técnicas de Machine Learning para el análisis de comorbilidades en pacientes diagnosticados con COVID-19 en Colombia. Inicialmente se hizo la construcción de una base de datos, a partir de la información de personas fallecidas diagnosticadas con COVID-19 reportada por el Ministerio de Salud y Protección Social de Colombia [3] y los informes diarios de fallecidos de la revista Semana [4]. Posteriormente se definieron las características más significativas, con un nivel de confianza del 95%, para clasificar los datos entre dos clases (riesgo alto y riesgo bajo de fallecimiento): edad, anemia, artritis, cáncer, cardiopatía, diabetes, enfermedad pulmonar obstructiva crónica, enfermedad renal, hipertensión, tabaquismo y obesidad (comorbilidades).

Luego se implementaron los modelos de aprendizaje supervisado de clasificación con las variables mencionadas anteriormente y se evaluaron los modelos con la precisión y la matriz de confusión. Se desarrollaron los modelos de k vecinos más cercanos, Bernoulli Naive Bayes, árbol de decisión, Gradient Boosting y máquina de soporte vectorial y las precisiones obtenidas fueron de 58.63%, 53.21%, 56.43%, 55.02% y 56.02% respectivamente. Debido a que la variación de las precisiones entre modelos es inferior al 5.5% se debe incluir nuevas variables relacionadas con la evolución del paciente antes del deceso para lograr definir los grupos creados a partir del etiquetado que se planteó en el presente proyecto.

Finalmente se desarrolló una aplicación web interactiva en la que es posible observar diferentes gráficas relacionadas con la base de datos construida, tales como las distribuciones por sexo y por edad, los días entre inicio de síntomas y muerte, el top de las comorbilidades reportadas y la cantidad de fallecidos por cada enfermedad, entre otras.

Palabras clave: COVID-19, comorbilidades, Machine Learning, aplicación web.

2 Introducción

El brote de enfermedad por Coronavirus 2019 (COVID-19) ha sido catalogado por la Organización Mundial de la Salud como una emergencia sanitaria para la salud pública de gran impacto y relevancia a nivel internacional. Desde el inicio de la pandemia, se han registrado en el mundo más de 35 millones de casos confirmados, de los cuales alrededor de 1 millón han fallecido, lo que corresponde aproximadamente al 3% del total de infectados [1]. La mayoría de las personas (alrededor del 80%) se recuperan de la enfermedad sin necesidad de tratamiento hospitalario. Sin embargo, alrededor de 1 de cada 5 personas que contraen la enfermedad provocada por el virus SARS-CoV-2, acaba presentando un cuadro grave, siendo más afectadas las personas que padecen afecciones médicas previas como hipertensión arterial, problemas cardiacos o pulmonares, diabetes o cáncer [2]. Aunque el porcentaje de infectados que presenta complicaciones es bajo, la rápida propagación del virus genera un aumento de la demanda al que se enfrentan los establecimientos sanitarios y los profesionales de la salud, lo que amenaza con sobrecargar algunos sistemas sanitarios e impedir su funcionamiento eficaz, trayendo consigo consecuencias catastróficas para la sociedad [5].

Actualmente, muchos estudios dirigen sus esfuerzos en comprender la relación que existe entre la muerte de pacientes con COVID-19 y enfermedades de base. Esto permitirá identificar adecuadamente la población de riesgo, teniendo en cuenta que las comorbilidades pueden influir en la vulnerabilidad de los pacientes o en el grado de afección de los pacientes diagnosticados con COVID-19. Como caso particular, se han empleado técnicas informáticas con alto impacto en el análisis de datos y predicción de modelos en diferentes áreas, como Machine Learning (ML) [6]. En el área de salud, en términos generales estos modelos han logrado predecir resultados en la atención médica, incluidos costo, utilización y calidad [7]. Por ejemplo, para predecir qué pacientes tienen más probabilidades de experimentar un reingreso hospitalario por insuficiencia cardíaca y para predecir los pacientes que pasan de un decil más bajo a uno más alto de los gastos de atención médica per cápita. Sin embargo, el aprendizaje automático sigue siendo un campo emergente y su aplicación a la investigación de resultados de atención médica está en crecimiento [7].

En el presente proyecto se realiza el análisis de las comorbilidades en pacientes diagnosticados con COVID-19 a través del uso de diferentes técnicas de Machine Learning, acordes con la naturaleza de los datos y las características de la población de interés. Es así como con los resultados obtenidos, se da cumplimiento a los ejes misionales del Alma Máter relacionados con el desarrollo científico y la investigación, para dar soluciones a las problemáticas de la sociedad y en especial del pueblo colombiano.

3 Objetivos

3.1 Objetivo General

Desarrollar un modelo mediante la aplicación de técnicas de Machine Learning para el análisis de comorbilidades en pacientes diagnosticados con SARS-CoV-2 en Colombia.

3.2 Objetivos específicos

1. Caracterizar la población de pacientes diagnosticados con SARS-CoV-2 de acuerdo a los datos publicados por el Ministerio de Salud y Protección Social de Colombia.
2. Construir una base de datos que permita realizar el entrenamiento y validación de los modelos a obtener a partir de las técnicas de Machine Learning.
3. Definir y evaluar el conjunto de técnicas de Machine Learning que aplican para el análisis de las comorbilidades de los pacientes diagnosticados (vivos y/o fallecidos).
4. Entrenar y validar los modelos seleccionados, considerando los modelos reportados en literatura que apliquen en el contexto del problema planteado.
5. Analizar la capacidad de predicción de los modelos de Machine Learning respecto a la realidad diaria de la pandemia en Colombia.

4 Marco Teórico

A continuación, se mencionan y describen los conceptos teóricos necesarios para el desarrollo del proyecto.

4.1 Coronavirus

Los coronavirus (CoV) son una extensa familia de virus que pueden causar diversas afecciones tanto en animales como en humanos. Se conoce que, en los humanos, varios coronavirus causan infecciones respiratorias que incluyen desde el resfriado común hasta enfermedades más graves como el Síndrome Respiratorio de Oriente Medio (MERS) y el Síndrome Respiratorio Agudo Severo (SRAS). Estas infecciones suelen causar fiebre y síntomas respiratorios (tos y disnea o dificultad para respirar) y en los casos más graves, pueden causar neumonía, síndrome respiratorio agudo severo, insuficiencia renal e, incluso, la muerte [6].

El COVID-19 es la enfermedad infecciosa causada por el coronavirus que se ha descubierto más recientemente (SARS-CoV-2). Tanto este nuevo virus como la enfermedad que provoca, eran desconocidos antes de que estallara el brote en Wuhan (China) en diciembre de 2019. Actualmente el COVID-19 es una pandemia que afecta a muchos países de todo el mundo, incluido Colombia [2]. La nueva infección por el virus SARS-CoV-2 se transmite principalmente por microgotas generadas en la vía aérea de una persona infectada y expulsadas al toser, estornudar y hablar. Las gotitas, que contienen viriones, pueden entrar en un huésped a través de las células epiteliales del tracto respiratorio superior. El virus utiliza una glicoproteína de doble dominio en su superficie (S1), que tiene una alta afinidad por los receptores de la enzima convertidora de angiotensina tipo 2 (ACE2), para invadir las células. Dada la amplia representación de estos receptores en los tejidos humanos, el virus en sí mismo tiene una gran capacidad para infectar varios tipos de células humanas e inducir diferentes cadenas patogénicas de eventos, que se corresponden con una variedad de cuadros clínicos de COVID-19 [8].

Inicialmente, la enfermedad COVID-19 comienza con síntomas de infección leve del tracto respiratorio como fiebre, tos y fatiga. Si bien los pacientes con COVID-19 presentan con mayor frecuencia una neumonía viral, algunos de ellos, progresan al síndrome de dificultad respiratoria aguda (SDRA) con

hipoxia profunda [8]. Como el SARS-CoV-2 es un virus nuevo en la población humana, existe una incertidumbre con respecto a los niveles virológicos en los pacientes y la relación con la gravedad de la enfermedad. Sin embargo, algunos estudios han informado una asociación entre cargas virales más altas y síntomas más graves. Uno de estos estudios (n = 76 pacientes) encontró que la carga viral media de los casos graves era alrededor de 60 veces mayor que la de casos leves (usando muestras nasofaríngeas), y esta relación se mantuvo desde las primeras hasta las últimas etapas de la infección. Además, el pico de la carga viral se encuentra aproximadamente durante la primera semana luego del inicio de síntomas y se vuelve indetectable en aproximadamente dos semanas [9]. Por tanto, es posible concluir que la carga viral de SARS-CoV-2 podría ser un marcador útil para evaluar la gravedad y el pronóstico de la enfermedad COVID-19 [10].

4.2 Comorbilidad de enfermedades

La comorbilidad es un término utilizado para describir la coexistencia de dos o más enfermedades en un mismo individuo, generalmente enfermedades correlacionadas. Estas enfermedades pueden ocurrir al mismo tiempo o una tras otra y esto también implica posibles interacciones entre las enfermedades que pueden empeorar el curso de ambas [11]. Las comorbilidades más comunes luego de padecer COVID-19 son la hipertensión arterial, diabetes y enfermedades cardiovasculares [12].

En términos de comorbilidades, las enfermedades cardiovasculares tienen la prevalencia más alta entre las enfermedades que ponen a los pacientes en mayor riesgo de amenaza del SARS-CoV-2, lo que puede ser explicado por la disminución de las citocinas proinflamatorias, que conduce a una función inmunológica más débil. También se ha encontrado que los fumadores son más susceptibles a las infecciones por coronavirus, especialmente a las especies más recientes. Sin embargo, no se han encontrado pruebas sólidas con respecto a la correlación de la enfermedad pulmonar obstructiva crónica (EPOC) y el tabaquismo con estar infectado con este nuevo virus [12].

4.3 Aprendizaje de Máquina

Del inglés, Machine Learning (ML), el aprendizaje de máquina es una técnica de inteligencia artificial (IA) que recrea la capacidad que tienen las máquinas

para "aprender", aportar soluciones a problemas concretos y generar conocimiento a partir de la información proporcionada a través de miles y millones de datos. El "cerebro" de esta técnica de aprendizaje artificial es el algoritmo, es decir, la serie de instrucciones que dan las pautas para estructurar los datos en forma de modelos, que permitan al sistema "pensar" y aportar soluciones o emitir un diagnóstico y/o predicción [6].

El proceso de aprendizaje comienza con observaciones o datos, como ejemplos, experiencia directa o instrucción, para buscar patrones en los datos y tomar decisiones en el futuro. El objetivo principal es permitir que las computadoras aprendan automáticamente sin intervención o asistencia humana y, en consecuencia, ajustar sus acciones. Los algoritmos de aprendizaje automático se clasifican como supervisados o no supervisados, sin embargo, también pueden existir clasificaciones adicionales como los semi-supervisados y los de refuerzo [13].

4.3.1 Algoritmos supervisados

Estos algoritmos aplican lo aprendido en el pasado a nuevos datos por medio de ejemplos etiquetados, para predecir eventos futuros. Se basa en el análisis de un conjunto de datos de entrenamiento conocido, para producir una función inferida y en consecuencia hacer predicciones sobre los valores de salida [13].

En el presente proyecto, se implementaron cinco algoritmos supervisados de aprendizaje automático de clasificación, los cuales se definen a continuación.

4.3.1.1 K vecinos más cercanos

El algoritmo de k vecinos más cercanos para la clasificación es posiblemente el más simple de todos los enfoques de clasificación supervisada. Las muestras de prueba simplemente se clasifican en la clase que ocurre con mayor frecuencia entre los k vecinos más cercanos en el espacio de parámetros multidimensionales. A pesar de su simplicidad, el método tiene una sólida base teórica en la estimación de densidad no paramétrica y, a menudo, puede superar a métodos mucho más sofisticados [14]. En la Figura 1 se muestra un ejemplo del funcionamiento del algoritmo, con un punto de prueba (azul) que se debe clasificar entre la clase verde y la clase roja.

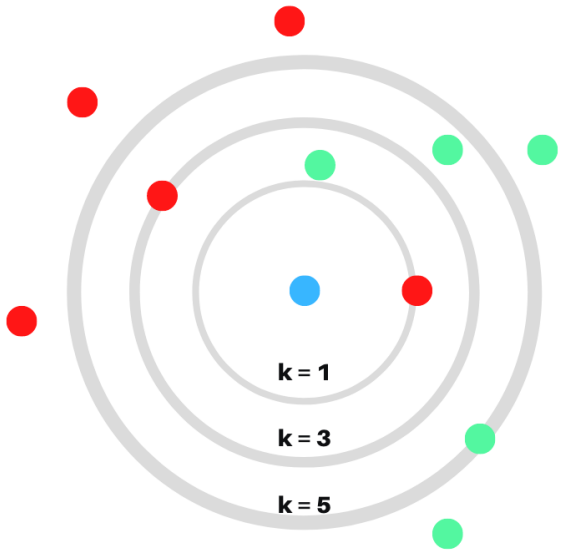

Figura 1. Ilustración de un modelo de clasificación de k vecinos más cercanos

El método requiere sólo la elección de k, que corresponde al número de vecinos a considerar al hacer la clasificación. Pequeños valores de k seleccionarán los puntos de entrenamiento más cercanos que sean más capaces de estimar la clasificación correcta en el punto de prueba. Sin embargo, debido al pequeño número, esta estimación será propensa a grandes fluctuaciones estadísticas. Por el contrario, los valores grandes de k reducen los errores estadísticos, pero permiten que los puntos lejanos contribuyan a la clasificación, lo que puede suavizar algunos de los detalles de las distribuciones de clases. Por lo general, se elige k como el valor que minimiza el error de clasificación en algunos datos de validación independientes o mediante procedimientos de validación cruzada [14].

4.3.1.2 Bernoulli Naive Bayes

Los algoritmos Naive Bayes son los algoritmos de clasificación estadística basados en el teorema de Bayes que ayudan a encontrar la probabilidad condicional de que sucedan dos eventos en función de las probabilidades de que ocurra cada evento individual. Estos algoritmos funcionan según el principio de que cada atributo es independiente del otro [15].

En particular, el algoritmo de Bernoulli Naive Bayes implementa los algoritmos de clasificación y entrenamiento de Bayes para los datos que se distribuyen

de acuerdo con distribuciones de Bernoulli multivariadas, es decir, puede haber varias características, pero se supone que cada una es una variable de valor binario. Por lo tanto, este algoritmo requiere que las muestras se representen como vectores de características con valores binarios. Si se le entrega cualquier otro tipo de datos, una instancia de Bernoulli Naive Bayes puede binarizar su entrada, dependiendo del parámetro a binarizar [15].

4.3.1.3 Árbol de decisión

Los árboles de decisión son herramientas utilizadas para la toma de decisiones y se asemejan a un diagrama de flujo que guía al lector a clasificar a una persona como de mayor riesgo o menor riesgo de un resultado. En un árbol de decisión, cada rama del árbol divide la población de estudio muestreada en subgrupos cada vez más pequeños que difieren en su probabilidad de un resultado de interés [7]. En la Figura 2 se muestra un ejemplo de un banco que utiliza un árbol de decisión para decidir si le debe ofrecer un préstamo a una persona, en este se muestra el nodo raíz, los nodos internos y el nodo terminal.

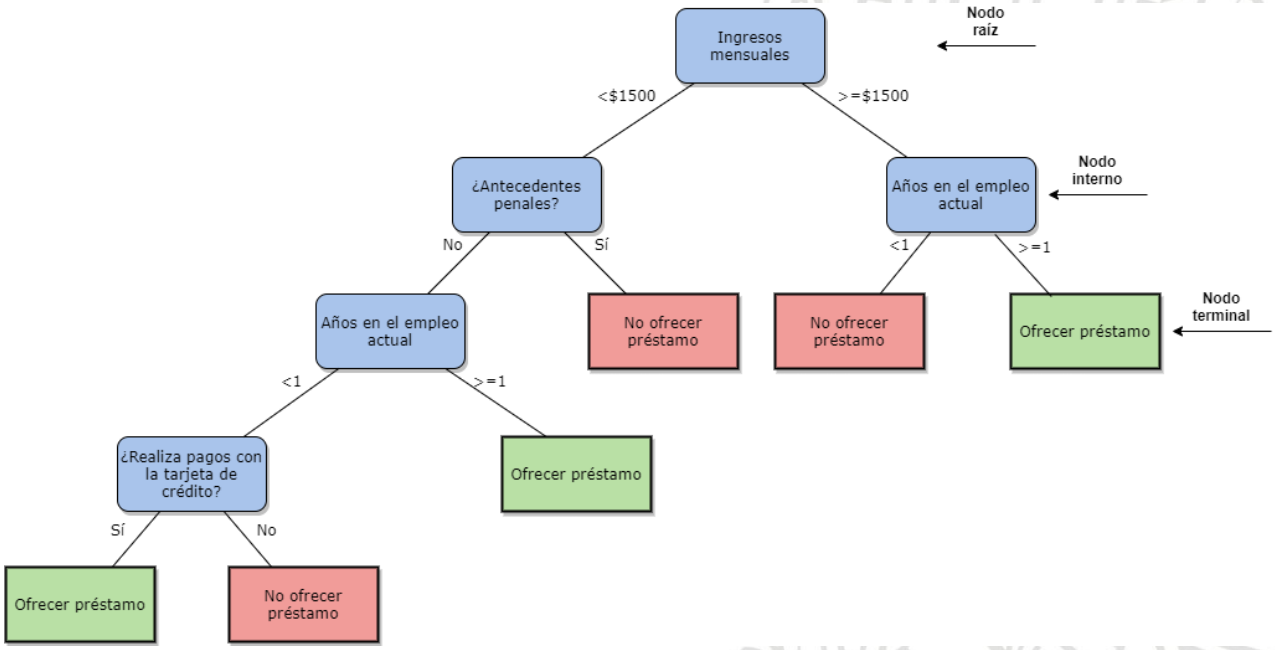

Figura 2. Ejemplo del algoritmo de árbol de decisión [16]

4.3.1.4 Gradient Boosting

El Gradient Boosting es un algoritmo que entrena secuencialmente a muchos árboles de decisión poco profundos para proporcionar una estimación más

precisa de la variable de respuesta. Cada nuevo árbol agregado al modelo de conjunto (combinación de todos los árboles anteriores) minimiza la función de pérdida asociada con el modelo de conjunto. La función de pérdida depende del tipo de tarea realizada y puede ser elegida por el usuario. Al agregar secuencialmente árboles que minimizan la función de pérdida (es decir, siguen el gradiente de la función de pérdida general), el error de predicción general disminuye [17].

Muchos hiperparámetros deben ajustarse para los árboles que aumentan el gradiente. Algunos de ellos controlan el proceso de aumento del gradiente, como la tasa de aprendizaje, la cantidad de árboles que se utilizarán, mientras que otros regulan el proceso de construcción de los árboles como tamaño mínimo de nodo, muestra del conjunto de datos que se utilizará y profundidad máxima [17].

4.3.1.5 Máquina de soporte vectorial

La máquina de soporte vectorial (SVM), es un clasificador que tiene como objetivo encontrar un hiperplano con un margen máximo para separar las clases de datos. El objetivo es maximizar el margen de este hiperplano, maximizando la distancia entre las muestras en su límite, estas muestras en el límite del hiperplano se denominan vectores de soporte. En casos no separables, es necesario agregar variables de holgura y un truco del Kernel para separar los datos. El truco del Kernel es una técnica comúnmente utilizada para resolver problemas linealmente inseparables y la función del Kernel realiza un mapeo no lineal en el espacio de características [18]. En la Figura 3 se muestra un ejemplo de una máquina de soporte vectorial con dos clases linealmente separables, en esta se muestran los vectores de soporte y el margen máximo.

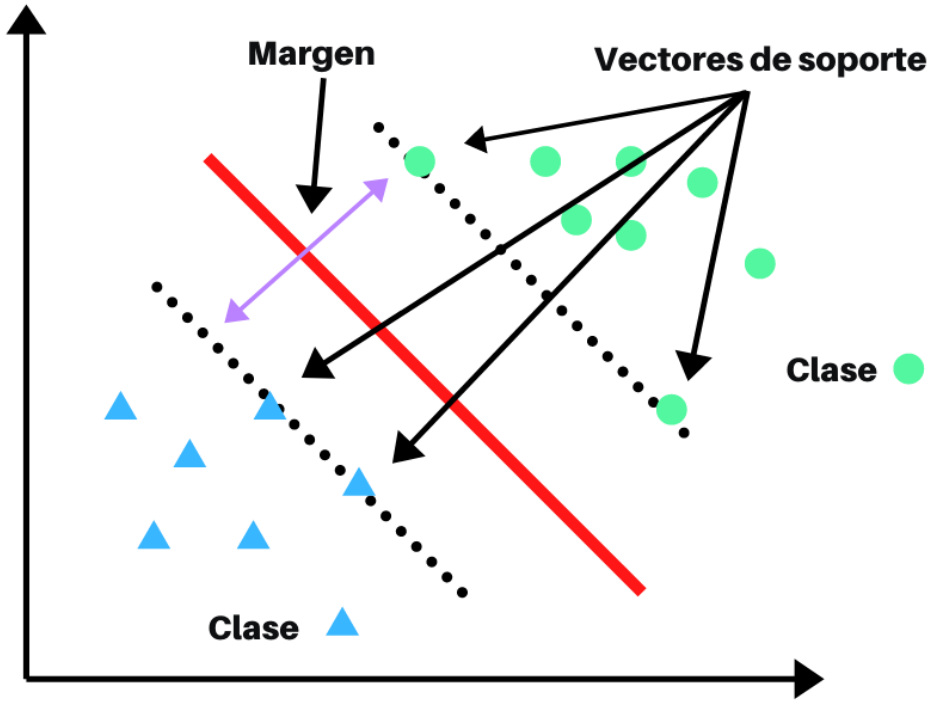

Figura 3. Ilustración de una máquina de soporte vectorial linealmente separable

4.3.2 Algoritmos no supervisados

Este tipo de algoritmo se utiliza cuando la información utilizada no está clasificada ni etiquetada. El algoritmo puede inferir una función para describir una estructura oculta y así, clasificar un conjunto de datos a través de elementos comunes entre ellos [19].

4.3.2.1 Método de clúster

El clustering trata de definir clases a partir de los datos sin conocer las etiquetas de clase. El propósito de estos algoritmos es identificar grupos de objetos, que son más similares entre sí que con otras agrupaciones. Esto con el fin de definir un conjunto de propiedades que pueda proporcionar una explicación intuitiva sobre aspectos relevantes de un conjunto de datos [20]. En la Figura 4 se muestra una ilustración del funcionamiento del algoritmo de clúster.

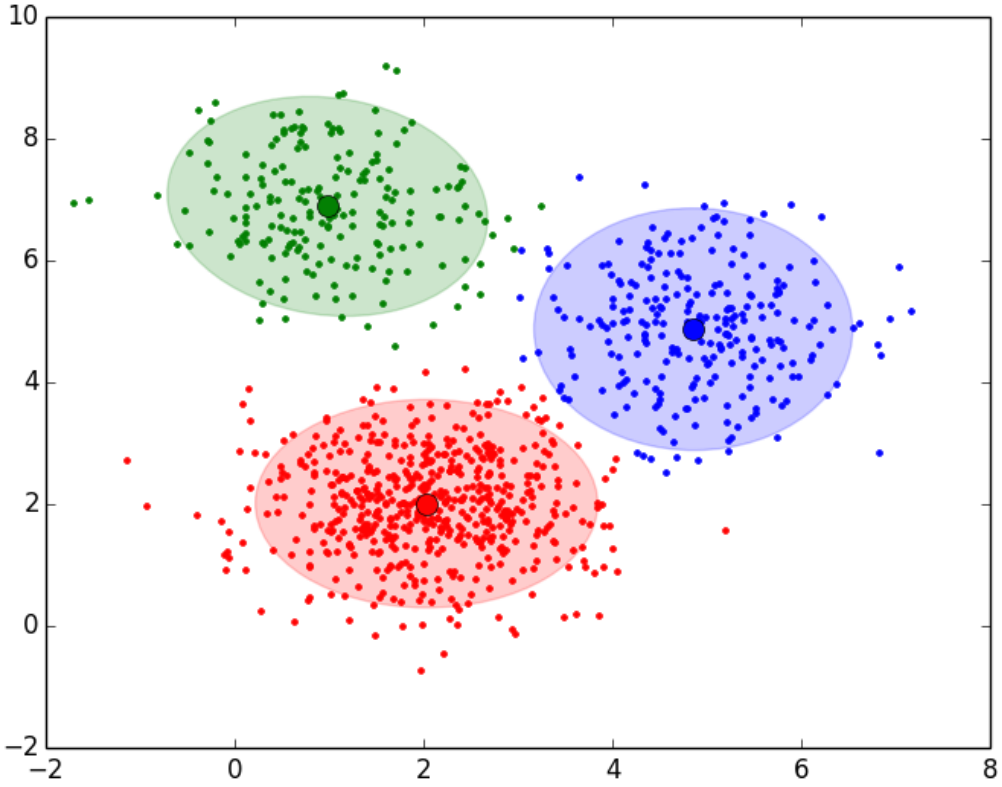

Figura 4. Ilustración del método de clúster [21]

4.3.2.2 Método de reducción de dimensionalidad

La reducción de dimensionalidad se refiere al proceso de mapear un punto n-dimensional, en un espacio k-dimensional más bajo. Esta operación reduce el tamaño para representar y almacenar un conjunto de datos, por lo que puede verse como un método para la compresión de datos [22].

4.3.3 Algoritmos semi-supervisados

Estos algoritmos utilizan datos etiquetados y no etiquetados para el entrenamiento. El objetivo es aprender un predictor que pueda mejorar la predicción de datos de pruebas futuras, con respecto al predictor aprendido sólo de los datos de entrenamiento etiquetados [23].

4.3.4 Algoritmos de refuerzo

Este método permite que las máquinas y los agentes de software determinen automáticamente el comportamiento ideal dentro de un contexto específico para maximizar su rendimiento [13].

5 Metodología

Para el desarrollo del proyecto se ejecutaron las etapas que se observan en la Figura 5 correspondiente a la metodología implementada. A continuación, se describen cada una de las actividades.

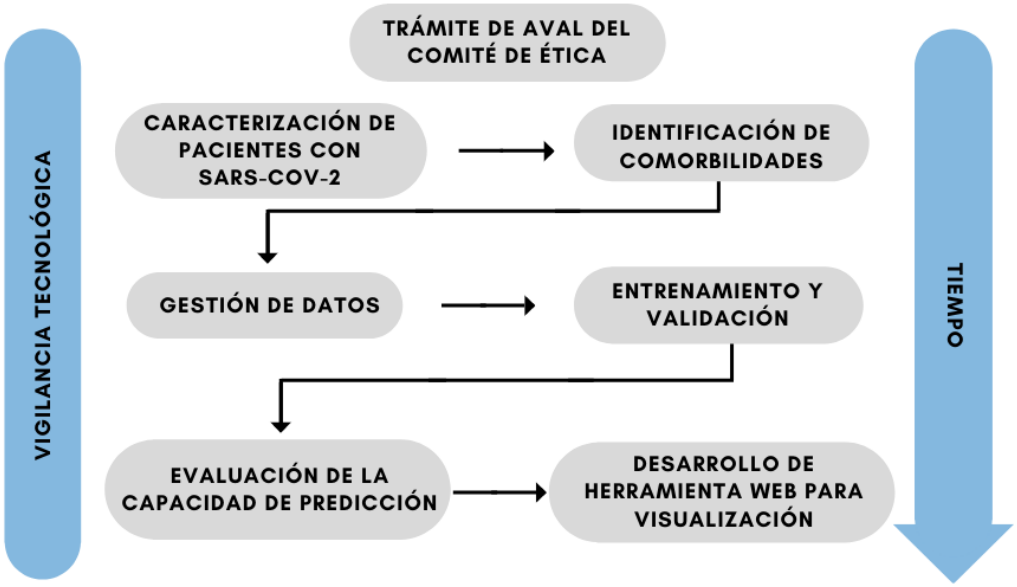

Figura 5. Metodología implementada para el desarrollo del proyecto

Vigilancia tecnológica

Durante el desarrollo del proyecto se realizó una permanente revisión en la literatura, incluyendo revistas y artículos científicos, bases de datos, entre otros, donde el principal propósito fue recopilar información acerca de las técnicas de Machine Learning más propicias para clasificar la población objetivo con los datos disponibles.

Caracterización de pacientes

La información obtenida a partir de los datos proporcionados por el Ministerio de Salud y Protección Social fue estandarizada con el objetivo de poder tener la variabilidad de la información clínica en el tiempo. Así mismo, a partir del

análisis de estos datos se realiza la identificación de las variables de interés como género, edad, ubicación y comorbilidades asociadas.

Identificación de comorbilidades

En esta fase, se realiza el análisis estadístico utilizando una prueba de hipótesis, para determinar cuáles de las comorbilidades y sus combinaciones aportan información diferente para discriminar los datos con respecto a un etiquetado.

Inicialmente, se determinó el etiquetado de los datos según el tiempo que permaneció cada caso entre el inicio de síntomas y la muerte. Se definieron dos etiquetas:

- Riesgo alto, en la cual se clasifican los pacientes que fallecieron entre los 0 y los 14 días desde el inicio de síntomas.
- Riesgo bajo, en la cual se clasifican los fallecidos que permanecieron 15 días o más entre el inicio de síntomas y la muerte.

Este etiquetado se puede observar en la Figura 6, donde se muestra que los datos son divididos en las dos clases, dependiendo del número de días que los fallecidos permanecieron entre el inicio de síntomas y la muerte.

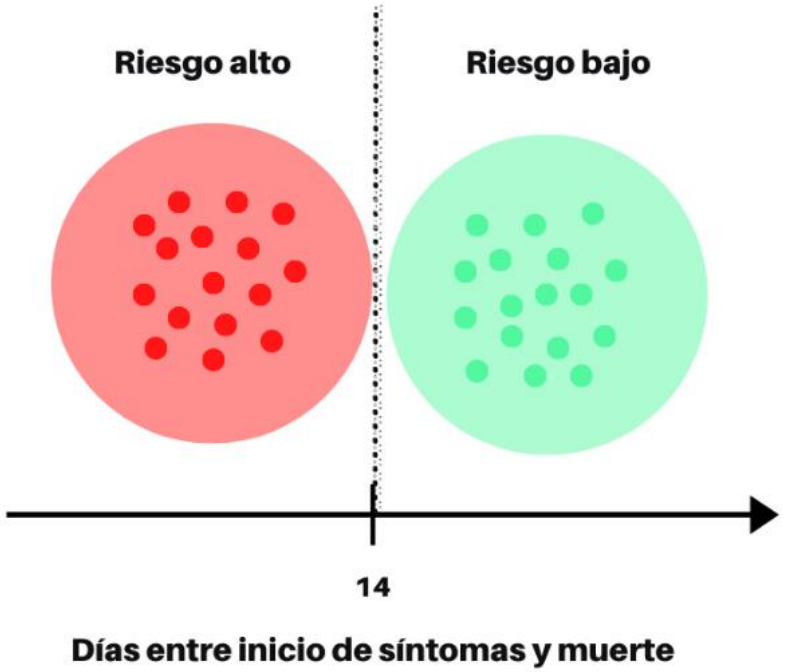

Figura 6. Esquema del etiquetado de los datos

Debido a que los datos de las comorbilidades corresponden a datos biológicos, se asume que los datos eran no paramétricos, por tanto, la prueba de hipótesis escogida fue la prueba de Kruskal-Wallis con un nivel de confianza del 95%. En esta prueba se tiene una hipótesis nula (H0) que plantea que todos los grupos provienen de la misma distribución y una hipótesis alternativa (H1) que dice que al menos uno de los grupos proviene de diferente distribución.

Gestión de datos

A partir de los datos suministrados por el Ministerio de Salud y Protección Social y de los reportes diarios de casos de COVID-19 de la Revista Semana, se alimentó la base de datos a utilizar, teniendo en cuenta las características más importantes de los pacientes, definidas anteriormente.

Entrenamiento y validación

Una vez definido el conjunto de datos a utilizar, en la fase de selección de la técnica de Machine Learning para el análisis de las comorbilidades en pacientes con COVID-19, se analizó la consistencia de los datos (número de variables y de muestras). Lo anterior con el objeto de no afectar la eficiencia

del modelo dado la posible alta dimensionalidad en el conjunto seleccionado.

El conjunto de datos fue subdivido de tal manera que se obtuvo un conjunto de datos de entrenamiento (80%) y un conjunto de datos de prueba (20%). Además, fue necesario balancear los datos, debido a que el número de datos de una clase duplicaba el número de datos de la otra clase y se implementaron cinco técnicas de Machine Learning supervisadas de clasificación que se enumeran a continuación con sus determinadas características:

I. Modelo de k vecinos más cercanos, se tunearon los parámetros de número de vecinos y función de peso.
II. Modelo de Bernoulli Naive Bayes, no se modificó ningún parámetro, por lo que se implementó con todos los valores predeterminados suministrados por la librería Scikit Learn.
III. Árbol de decisión se fijaron el mínimo número de muestras para dividir un nodo interno y el mínimo número de muestras para estar en un nodo hoja.
IV. Modelo de Gradient Boosting se implementó con diferentes valores de estimadores.
V. Máquina de soporte vectorial se implementó con diferentes parámetros de regularización y diferentes Kernel.

Evaluación del modelo

Los datos publicados diariamente por el Ministerio de Salud y Protección Social permitieron evaluar la capacidad de predicción del modelo con un nuevo conjunto de datos una vez estos sean estandarizados. Lo anterior es acorde con el escenario de que esta pandemia no se solucionará en el corto plazo, por lo cual la disponibilidad de nueva información será continua.

Desarrollo de herramienta web

Los datos utilizados pueden ser visualizados de forma abierta por la comunidad en general, desde una herramienta web desarrollada en el contexto de este trabajo de investigación. En la cual, se pueden modificar diferentes variables como la ubicación de los casos, meses a visualizar, etc.

para que los usuarios puedan interactuar y observar la información que sea de su interés. Toda la información publicada guardando la confidencialidad.

Criterios éticos

De acuerdo a la resolución 8430 de 1993 (octubre 4) Por la cual se establecen las normas científicas, técnicas y administrativas para la investigación en salud. Artículo 11. Se considera Investigación sin riesgo: Son estudios que emplean técnicas y métodos de investigación documental retrospectiva y aquellos en los que no se realiza ninguna intervención o modificación intencionada de las variables biológicas, fisiológicas, psicológicas o sociales de los individuos que participan en el estudio.

Ley estatutaria 1581 de 2012. Por la cual se dictan disposiciones generales para la protección de datos personales. Cumplimiento del principio de seguridad: la información sujeta a tratamiento se deberá manejar con las medidas técnicas, humanas y administrativas que sean necesarias para otorgar seguridad a los registros evitando su adulteración, pérdida, consulta, uso o acceso no autorizado o fraudulento.

Artículo 5. Excepciones para el tratamiento de datos sensibles: cuando el tratamiento tenga una finalidad histórica, estadística o científica. En este evento deberán adoptarse las medidas conducentes a la supresión de identidad de los titulares.

Por lo tanto, este trabajo de investigación se considera una investigación sin riesgos y todos los datos serán utilizados solo para fines académicos.

6 Resultados y análisis

Posterior a la implementación de la metodología mencionada anteriormente, fueron obtenidos los siguientes resultados, los cuales están enumerados y descritos a continuación.

6.1 Caracterización de pacientes diagnosticados con SARS-CoV-2

Debido a que los datos suministrados por el Ministerio de Salud y Protección Social de Colombia no cuentan con la información relacionada a las comorbilidades de todos los casos diagnosticados con SARS-CoV-2, fue necesario utilizar adicionalmente los datos suministrados por el reporte diario de fallecidos por COVID-19 de la revista Semana, el cual si contenía información de las comorbilidades de los fallecidos. Por tanto, para el presente proyecto, sólo se tuvieron en cuenta datos de las personas fallecidas a causa del virus en Colombia.

Una vez cruzada la información de las dos fuentes, fue necesario definir las variables a tener en cuenta para la creación de la nueva base de datos. En la Tabla 1 se muestran las variables con su definición.

Tabla 1. Variables de la base de datos

Variable	Tipo	Definición
ID del caso	Identificadora	Identificación del caso
Número de caso	Identificadora	Número del caso registrado en los boletines de Casos COVID-19 Colombia en la página web del Instituto Nacional de Salud
Departamento	Demográfica	Departamento de ubicación del caso
Ciudad	Demográfica	Ciudad de ubicación del caso
Edad	Demográfica	Edad en años del fallecido con COVID-19 confirmado

Variable	Tipo	Definición
Sexo	Demográfica	Sexo del fallecido con COVID-19 confirmado (M: Masculino, F: Femenino)
Fecha de inicio de síntomas	Diagnóstico	Fecha en la cual el fallecido reportó haber iniciado síntomas (DD/MM/AAAA)
Fecha de muerte	Diagnóstico	Fecha de fallecimiento reportada del paciente (DD/MM/AAAA)
Fecha de diagnóstico	Diagnóstico	Fecha en la cual se confirmó el diagnóstico por laboratorio (DD/MM/AAAA)
Días entre inicio de síntomas y muerte	Diagnóstico	Días entre la fecha de inicio de síntomas y la fecha de fallecimiento del paciente
Días entre inicio de síntomas y diagnóstico	Diagnóstico	Días entre la fecha de inicio de síntomas y la fecha de confirmación del diagnóstico del paciente
Comorbilidades	Clínica	Una columna por cada comorbilidad registrada. La variable es 1 si el paciente presenta la enfermedad o 0 si el paciente no la presenta

6.2 Construcción de la base de datos

A partir de los datos suministrados por el Ministerio de Salud y Protección Social de Colombia y por los reportes diarios de la Revista Semana, se alimentó la base de datos con las variables mencionadas anteriormente. Sin embargo, a medida que se fueron ingresando más datos, el número de características fue aumentando significativamente, debido a la gran cantidad de comorbilidades que fueron registradas. En esta nueva base de datos se guardó la información de 5622 fallecidos, de los cuales sólo fueron utilizados 3553 debido a que fue necesario eliminar los casos en los cuales se reportaron las comorbilidades en estudio.

Por esto, se realizó la prueba de Kruskall-Wallis, para determinar cuáles de las comorbilidades y sus combinaciones aportaban información diferente para discriminar los datos entre riesgo alto y riesgo bajo. Con un nivel de confianza del 95%, se obtuvieron las combinaciones de comorbilidades que se muestran en la Figura 7, con las cuales se obtuvo un valor p menor a 0.05.

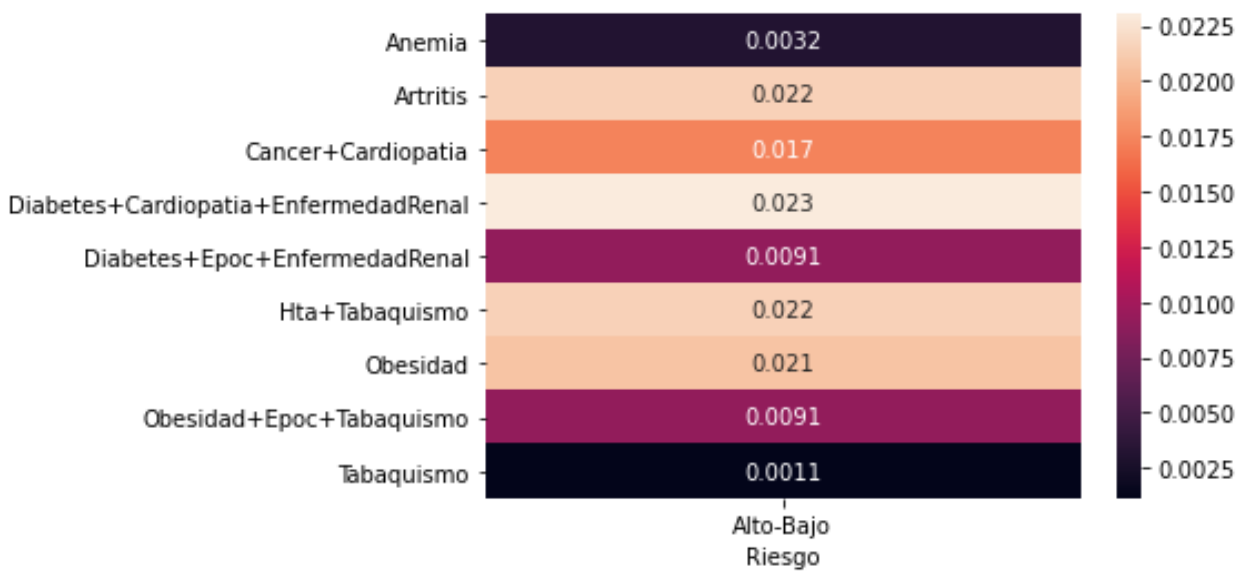

Figura 7. Valores p para las combinaciones de comorbilidades. Se muestran los valores p en una escala de colores para observar las comorbilidades más significativas con un 95% de confianza.

Como se puede observar en la Figura 7, se obtuvieron nueve combinaciones de comorbilidades que permiten diferenciar entre el grupo de riesgo alto y el grupo de riesgo bajo con un nivel de confianza del 95%. Por tanto, fue posible determinar las características de la base de datos utilizada para el entrenamiento y validación de los modelos de Machine Learning. Estas características fueron las enfermedades: anemia, artritis, cáncer, cardiopatía, diabetes, EPOC, enfermedad renal, hipertensión, tabaquismo y obesidad, además de la edad que también resultó ser significativa para diferenciar entre las dos clases.

Así, de la base de datos inicial, en la cual se tenían un total de 125 comorbilidades, sólo 11 características describen la población de acuerdo a la variable de clasificación grupal. Dentro de éstas se encuentran las 10 comorbilidades más significativas, las cuales representan aproximadamente el 87% del total de los datos.

6.3 Definición de las técnicas de Machine Learning.

Para la elección de las técnicas de Machine Learning que se utilizaron, se tuvo en cuenta varios aspectos: (1) el tipo de problema a abordar, es decir la clasificación de pacientes, por lo que se escogieron técnicas de clasificación supervisada; (2) la cantidad de datos disponible; (3) el número de características a tener en cuenta y (4) los algoritmos de clasificación disponibles en la librería Scikit Learn: k vecinos más cercanos, Bernoulli Naive Bayes, árbol de decisión, Gradient Boosting y máquina de soporte vectorial.

6.4 Entrenamiento y validación de los modelos

A continuación, se enumeran los cinco modelos implementados y se describen los resultados obtenidos para cada uno de ellos. Es importante mencionar que, para todos los modelos fue necesario balancear los datos de las dos clases, obteniendo 1243 datos por clase, para un total de 2486 datos utilizados en total, tal y como se muestra en la Figura 8.

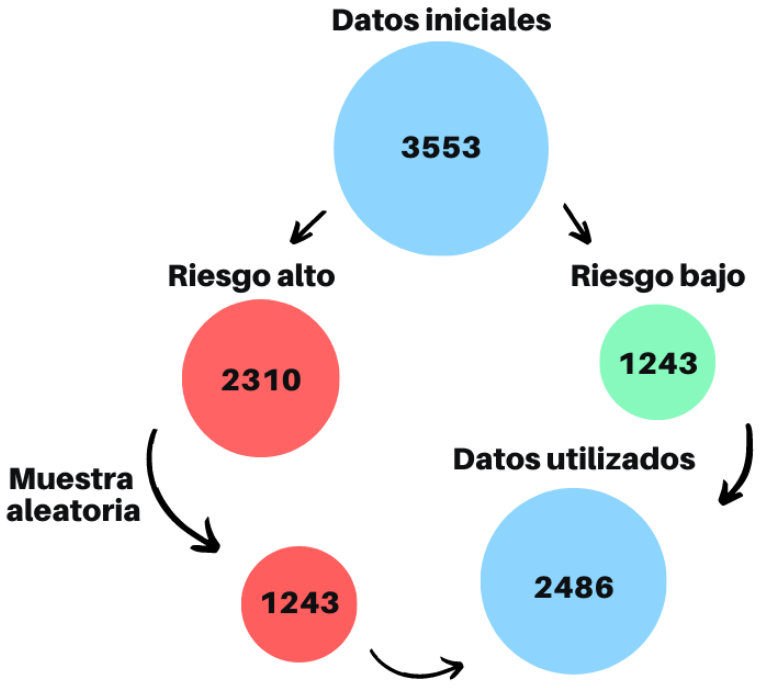

Figura 8. Esquema de cómo se obtuvieron los datos utilizados en los modelos

6.4.1 K vecinos cercanos

Inicialmente se implementaron modelos con diferente número de vecinos para determinar el valor de k con el que se obtenía la mayor precisión (Figura 9 y 10).

```
from sklearn import metrics
#Se grafica la precisión vs el número de vecinos
acc = []
for i in range(1,100):
    #Se crea el modelo y se entrena para k desde 1 hasta 100
    neigh = KNeighborsClassifier(n_neighbors = i).fit(X_train,y_train)
    #Se genera la predicción
    yhat = neigh.predict(X_test)
    #Se guarda la precisión
    acc.append(metrics.accuracy_score(y_test, yhat))

plt.figure(figsize=(10,6))
plt.plot(range(1,100),acc,color = 'darkred',linestyle='dashed',
         marker='o',markerfacecolor='skyblue', markersize=7)
plt.title('Precisión vs Número de vecinos')
plt.xlabel('K')
plt.ylabel('Precisión')
plt.savefig('k_values.png',bbox_inches='tight')
print("Maximum accuracy:",max(acc),"at K =",acc.index(max(acc)))
```

Figura 9. Código para graficar k vs precisión

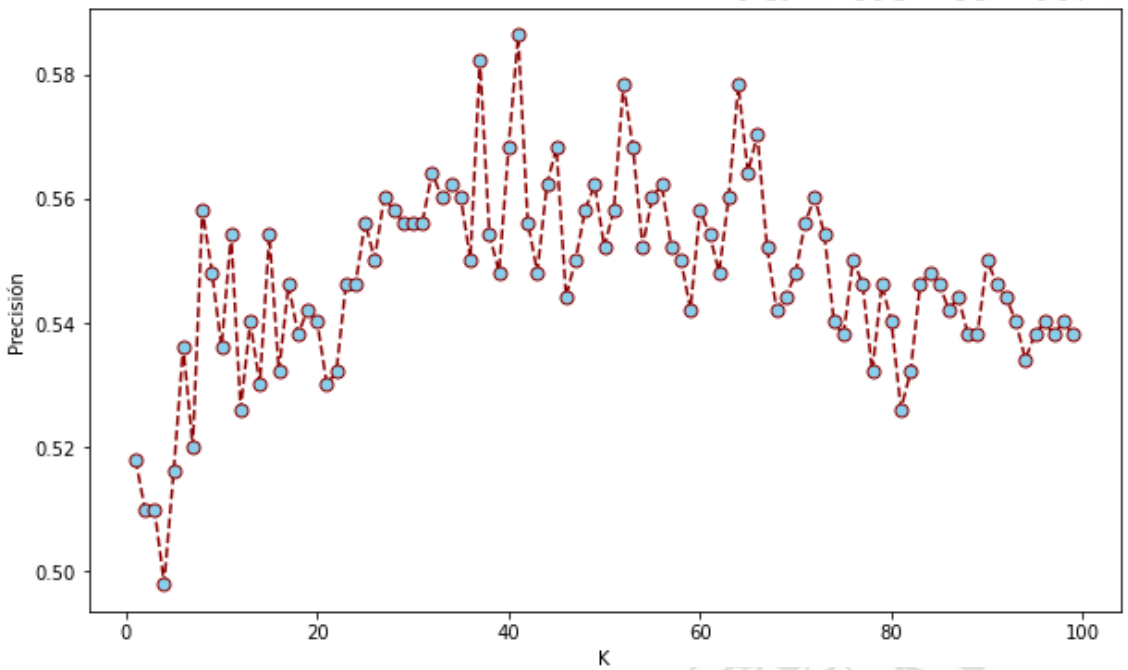

Figura 10. Precisión vs número de vecinos

Como se observa en la Figura 10, la precisión del clasificador de k vecinos más cercanos comienza aumentando hasta llegar a su punto máximo y luego empieza a descender. Es posible determinar que la mayor precisión (58.63%), se encuentra con un número de vecinos k igual a 40. Por tanto, se obtuvo la matriz de confusión (Figura 11) para el modelo con 40 vecinos y con dos funciones de peso diferentes (uniforme y distancia) y se muestran en la Figura 12.

```
import matplotlib.pyplot as plt
from sklearn.metrics import plot_confusion_matrix

#Se define K=40 por ser el número de vecinos con el que se obtuvo mayor precisión
k=40
#Se define el modelo con función de peso uniforme
knn = KNeighborsClassifier(n_neighbors=k,weights='uniform',algorithm='auto')
#Se entrena el modelo
knn.fit(X_train, y_train)
#Se define el modelo con función de peso distancia
knn2 = KNeighborsClassifier(n_neighbors=k,weights='distance',algorithm='auto')
#Se entrena el modelo
knn2.fit(X_train, y_train)

#Se grafican las dos matrices de confusión
classifiers=[knn,knn2]
fig, axs = plt.subplots(1, 2, figsize=(15,10))
for cls,ax in zip(classifiers, axs.flatten()):
    if cls==knn:
        plot_confusion_matrix(cls,X_test, y_test,ax=ax,cmap=plt.cm.Blues)
        ax.title.set_text('Weights=Uniform K= '+str(k)+' Precisión= '+str(round(knn.score(X_test, y_test),2)))
    elif cls==knn2:
        plot_confusion_matrix(cls,X_test, y_test,ax=ax,cmap=plt.cm.Reds)
        ax.title.set_text('Weights=Distance K= '+str(k)+' Precisión= '+str(round(knn2.score(X_test, y_test),2)))
```

Figura 11.Código para obtener las matrices de confusión de k vecinos más cercanos

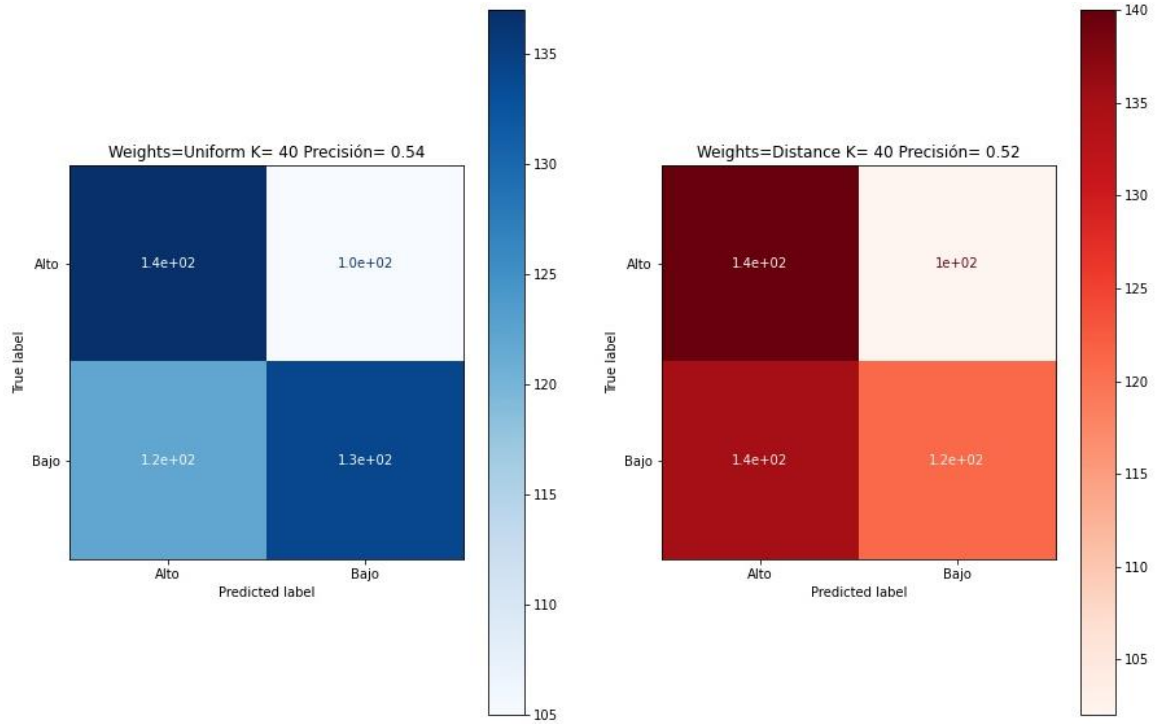

Figura 12. Matrices de confusión para los modelos con 40 vecinos funciones de peso uniforme y distancia

En la Figura 12 se observa principalmente que no existe una diferencia significativa entre los resultados de las dos funciones de peso, sin embargo, con la función de peso uniforme se obtuvo una mayor precisión. Es posible notar que el modelo de k vecinos más cercanos es bueno clasificando la clase de riesgo alto, pero comete más errores clasificando la clase de riesgo bajo.

Lo anterior se evidencia al realizar validación cruzada con 10 pliegues, cuyas precisiones obtenidas no superan el 61% (Tabla 2).

Tabla 2. Resultado de la validación cruzada para el modelo de k vecinos más cercanos

Modelo	Precisión
1	56.63%
2	56.22%
3	54.22%
4	60.64%
5	54.62%
6	57.03%
7	54.84%

Modelo	Precisión
8	48.79%
9	57.66%
10	54.03%

Como se puede notar en la Tabla 2, al realizar la validación cruzada del modelo con 40 vecinos y función de peso uniforme, las precisiones no tienen un cambio significativo, por lo que es posible decir que el modelo tiene un error de varianza bajo, debido a que el resultado no varía al cambiar los datos de entrenamiento. Sin embargo, el modelo si presenta un error de bias muy alto.

Por tanto, es posible decir que, para la base de datos obtenida, este clasificador no es viable debido a que la exactitud obtenida es demasiado baja y esto puede deberse a la naturaleza de los datos (dicotómicos).

6.4.2 Bernoulli Naive Bayes

Para la implementación del modelo de Bernoulli Naive Bayes, no se modificó ningún parámetro y se utilizó el modelo con todos los valores predeterminados por la librería Scikit Learn (Figura 13). Luego de entrenar el modelo se obtuvo una precisión de 53.21% y la matriz de confusión mostrada en la Figura 14.

```
from sklearn.metrics import plot_confusion_matrix
from sklearn.naive_bayes import BernoulliNB

#Se crea el modelo
nbb = BernoulliNB()
#Se entrena el modelo
nbb.fit(X_train, y_train)
#Se obtiene la precisión del modelo
score=nbb.score(X_test, y_test)
#Se genera la matriz de confusión
plot_confusion_matrix(nbb,X_test, y_test,labels=['Alto','Bajo'],cmap=plt.cm.Blues)
```

Figura 13. Código para generar la matriz de confusión de Bernoulli Naive Bayes

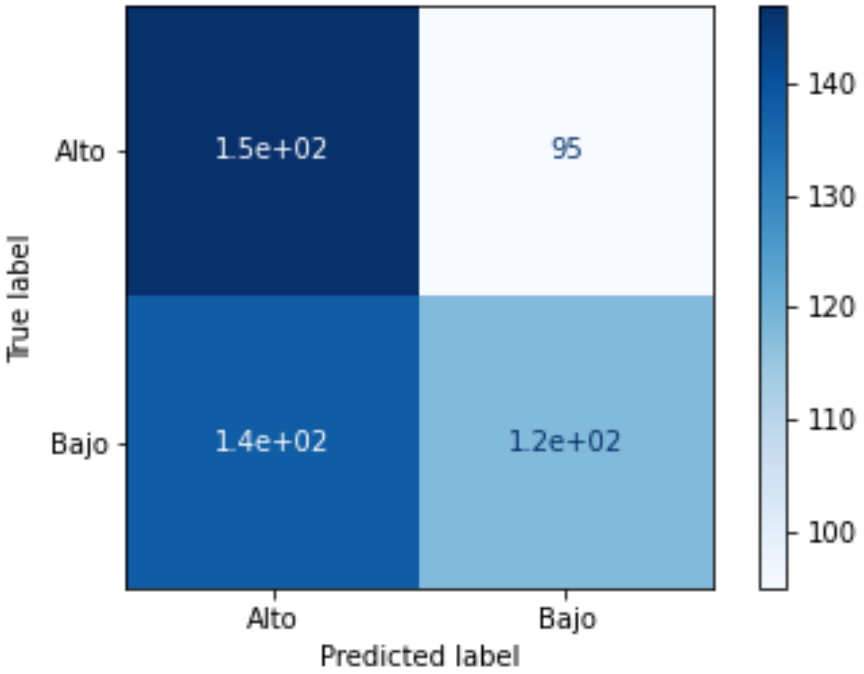

Figura 14. Matriz de confusión para el modelo de Bernoulli Naive Bayes

Es posible observar en la Figura 14, que el modelo de Bernoulli Naive Bayes, al igual que el modelo de k vecinos más cercanos, tiene un mejor desempeño clasificando la clase de riesgo alto que clasificando la clase de riesgo bajo, sin embargo, con este modelo se obtuvo una precisión más baja que con el modelo anterior.

Además, se realizó una validación cruzada del modelo con 10 pliegues y las precisiones obtenidas se muestran en la Tabla 3.

Tabla 3. Resultado de la validación cruzada para el modelo de Bernoulli Naive Bayes

Modelo	Precisión
1	53.41%
2	51.41%
3	54.62%
4	51.81%
5	51.00%
6	51.41%
7	49.60%

Modelo	Precisión
8	50.40%
9	49.19%
10	50.81%

Como se muestra en la Tabla 3, las precisiones del modelo no varían significativamente al realizar la validación cruzada, por lo que este modelo, también tiene un error de varianza bajo pero un error de bias muy alto. En consecuencia, se puede decir que este clasificador no es adecuado para la base de datos en cuestión debido a que se obtuvo un rendimiento muy bajo.

6.4.3 Árbol de decisión

En el modelo de árbol de decisión se modificaron dos parámetros, el mínimo número de muestras para dividir un nodo interno (min_samples_split) que se fijó en 400 y el mínimo número de muestras para dejar en un nodo hoja (min_samples_leaf) que se fijó en 100. Luego de entrenar el modelo (Figura 15), se obtuvo una precisión del 56.43% y la matriz de confusión mostrada en la Figura 16.

```
from sklearn import tree
from sklearn.metrics import plot_confusion_matrix

#Se crea el modelo
tree = tree.DecisionTreeClassifier(min_samples_split=400,min_samples_leaf=100)
#Se entrena el modelo
tree.fit(X_train, y_train)
#Se obtiene la precisión
score=tree.score(X_test, y_test)
#Se obtiene la matriz de confusión
plot_confusion_matrix(tree,X_test, y_test,labels=['Alto','Bajo'],cmap=plt.cm.Blues)
```

Figura 15. Código para generar la matriz de confusión del árbol de decisión

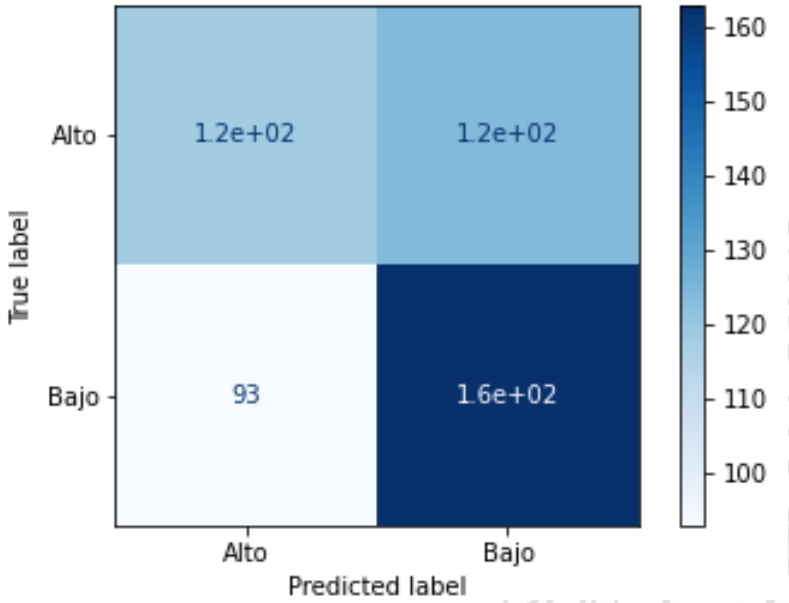

Figura 16. Matriz de confusión para el modelo de árbol de decisión

Se puede observar en la Figura 16 que el clasificador de árbol de decisión clasifica de mejor manera la clase de riesgo bajo a comparación de la clase de riesgo alto, debido a que, en esta última, clasifica adecuadamente sólo la mitad de los datos. Además, se realizó una validación cruzada del modelo con 10 pliegues y las precisiones obtenidas se muestran en la Tabla 4.

Tabla 4. Resultado de la validación cruzada para el modelo de árbol de decisión

Modelo	Precisión
1	57.83%
2	57.03%
3	53.82%
4	51.41%
5	58.63%
6	57.03%
7	58.47%
8	58.87%
9	52.82%
10	58.87%

Según las precisiones mostradas en la Tabla 4, también es posible mencionar que para este modelo el error de varianza es muy bajo y el error de bias es muy alto. Por tanto, el modelo implementado de árbol de decisión con los parámetros definidos no tuvo un buen desempeño, por lo que es posible considerar tunear otros parámetros diferentes para mejorar la predicción.

6.4.4 Gradient Boosting

Para el modelo de Gradient Boosting fue necesario definir inicialmente el número de estimadores que generaba la mayor precisión (Figura 17). Para esto, se implementaron modelos con diferente número de estimadores y se obtuvo la Figura 18.

```
#Se grafica la precisión vs el número de estimadores
acc = []
from sklearn import metrics
for i in range(1,100):
    #Se crea el modelo y se entrena un número de estimadores desde 1 hasta 100
    gbc = GradientBoostingClassifier(n_estimators=i).fit(X_train,y_train)
    #Se genera la predicción
    yhat = gbc.predict(X_test)
    #Se guarda la precisión
    acc.append(metrics.accuracy_score(y_test, yhat))

plt.figure(figsize=(10,6))
plt.plot(range(1,100),acc,color = 'darkred',linestyle='dashed',
         marker='o',markerfacecolor='skyblue', markersize=10)
plt.title('Precisión vs Número de estimadores')
plt.xlabel('Estimadores')
plt.ylabel('Precisión')
plt.savefig('estimators_boosting.png',bbox_inches='tight')
print("Maximum accuracy:",max(acc),"at estimators =",acc.index(max(acc)))
```

Figura 17.Código para graficar número de estimadores vs precisión

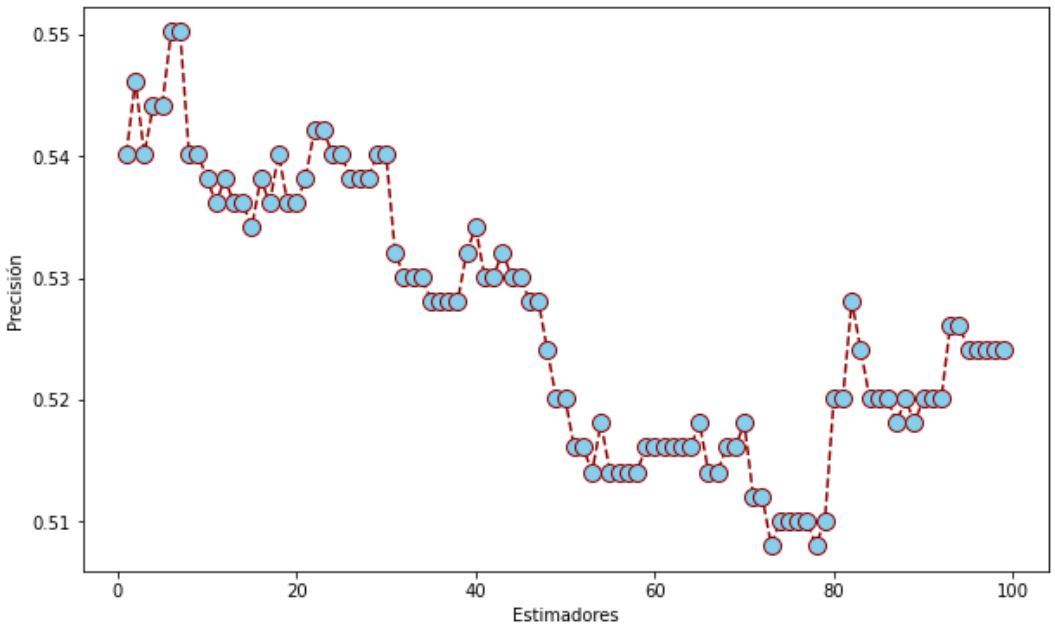

Figura 18. Precisión vs número de estimadores para el modelo de Gradient Boosting

De la Figura 18 se puede determinar que el mayor valor de precisión para el modelo de Gradient Boosting, se encuentra con 5 estimadores y es de 55.02%. Por tanto, se obtuvo la matriz de confusión (Figura 19) para el modelo con este número de estimadores y se muestra en la Figura 20.

```
#Se genera el modelo con 5 estimadores
gbc = GradientBoostingClassifier(n_estimators=5)
#Se entrena el modelo
gbc.fit(X_train, y_train)
#Se grafica la matriz de confusión
plot_confusion_matrix(gbc,X_test, y_test,labels=['Alto','Bajo'],cmap=plt.cm.Blues)
```

Figura 19. Código para graficar la matriz de confusión del modelo Gradient Boosting

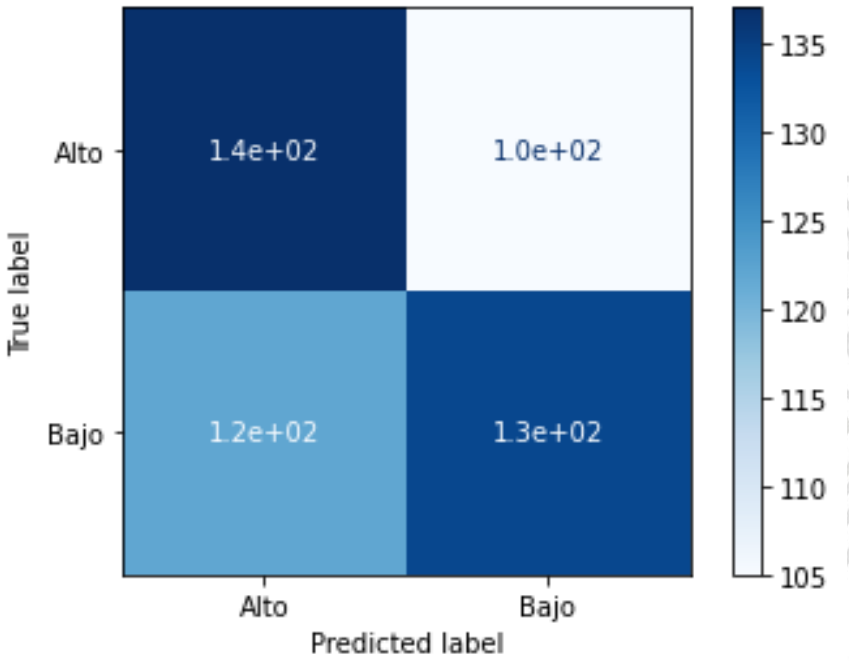

Figura 20. Matriz de confusión para el modelo de Gradient Boosting

Según los resultados mostrados en la Figura 20, es posible decir que con este modelo no se obtuvo un buen desempeño, debido a que una gran cantidad de datos fueron clasificados en la clase equivocada. Por esto, también se realizó una validación cruzada del modelo con 10 pliegues y las precisiones obtenidas se muestran en la Tabla 5.

Tabla 5. Resultado de la validación cruzada para el modelo de Gradient Boosting

Modelo	Precisión
1	58.63%
2	58.23%
3	56.63%
4	54.22%
5	53.82%
6	53.82%
7	52.42%
8	55.24%
9	54.43%

Modelo	Precisión
10	49.60%

Como se observa en la Tabla 5, las precisiones no tienen una variación significativa durante la validación cruzada, por lo que este modelo tiene un bajo error de varianza y un alto error de bias, el cual podría ser reducido en este modelo con la variación de algunos otros parámetros como la función de pérdida, la taza de aprendizaje, etc.

6.4.5 Máquina de soporte vectorial

En la implementación de la máquina de soporte vectorial se tuvieron en cuenta dos parámetros, el parámetro de regularización (C) y el kernel. Inicialmente con un kernel lineal se definieron los modelos con diferentes valores de C y se obtuvo la Figura 21.

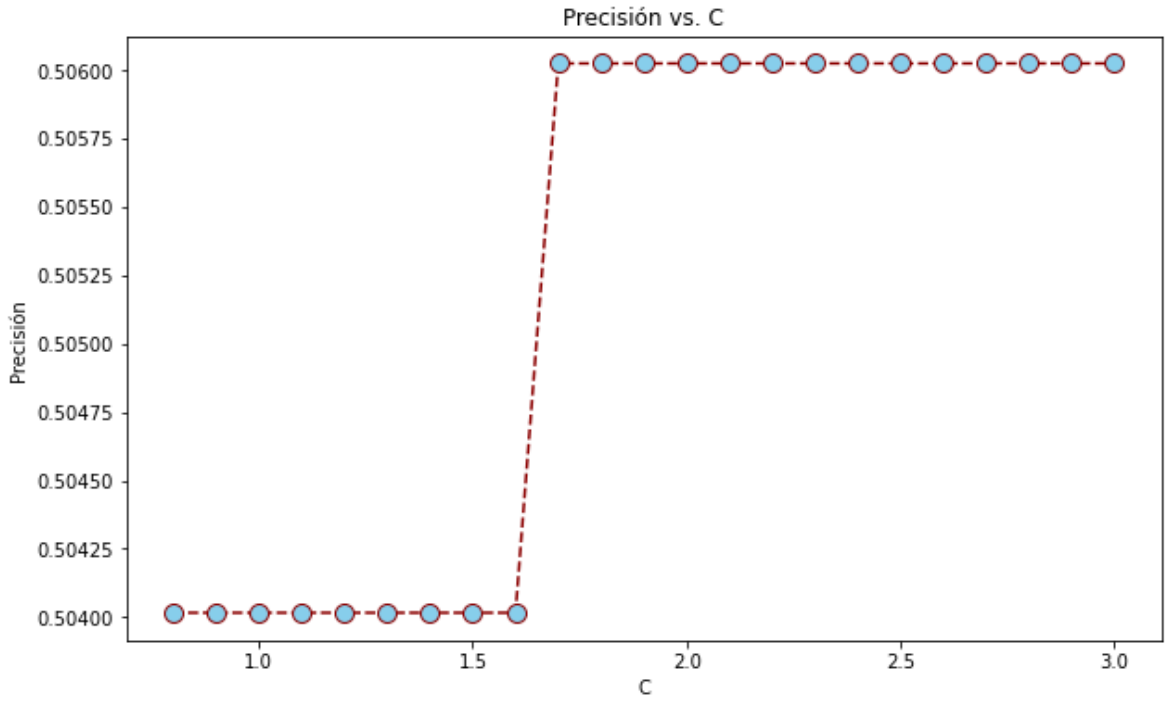

Figura 21. Precisión vs valor C para la máquina de soporte vectorial con kernel lineal

De la Figura 21 se puede determinar que la máxima precisión alcanzada por el modelo con kernel lineal es de 50.60% con un parámetro de regularización de 1.7. Por tanto, se obtuvo la matriz de confusión para el modelo con estas características (Figura 22) y se muestra en la Figura 23.

```
#Se crea el modelo de kernel lineal y c=1.7
svm=SVC(C=1.7,kernel='linear')
#Se entrena el modelo
svm.fit(X_train, y_train)
#Se grafica la matriz de confusión
plot_confusion_matrix(svm,X_test, y_test,labels=['Alto','Bajo'],cmap=plt.cm.Blues)
```

Figura 22. Código para generar la matriz de confusión de la máquina de soporte vectorial con kernel lineal

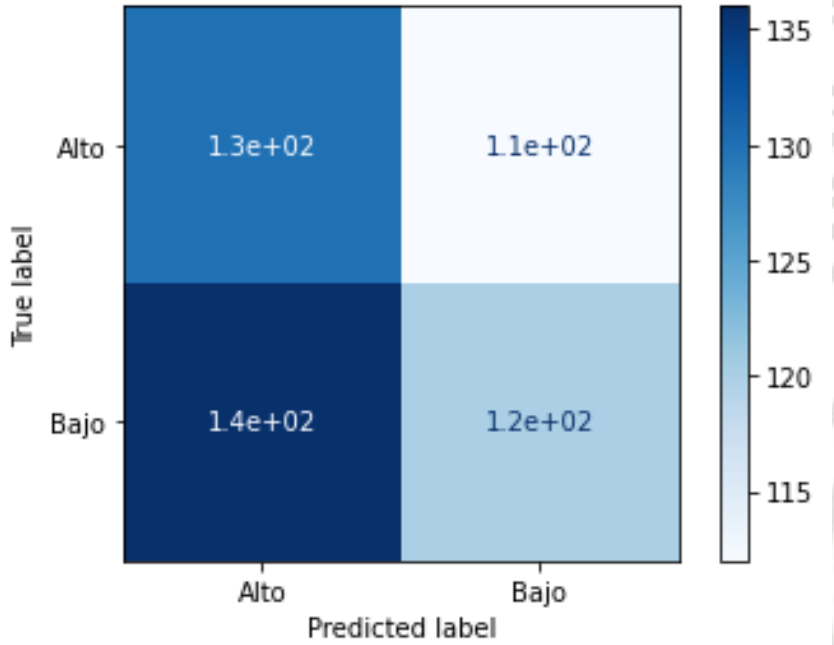

Figura 23. Matriz de confusión para la máquina de soporte vectorial con Kernel lineal

También, se definieron modelos con kernel de función de base radial (RBF) y diferentes valores del parámetro de regularización y la gráfica obtenida se muestra en la Figura 24.

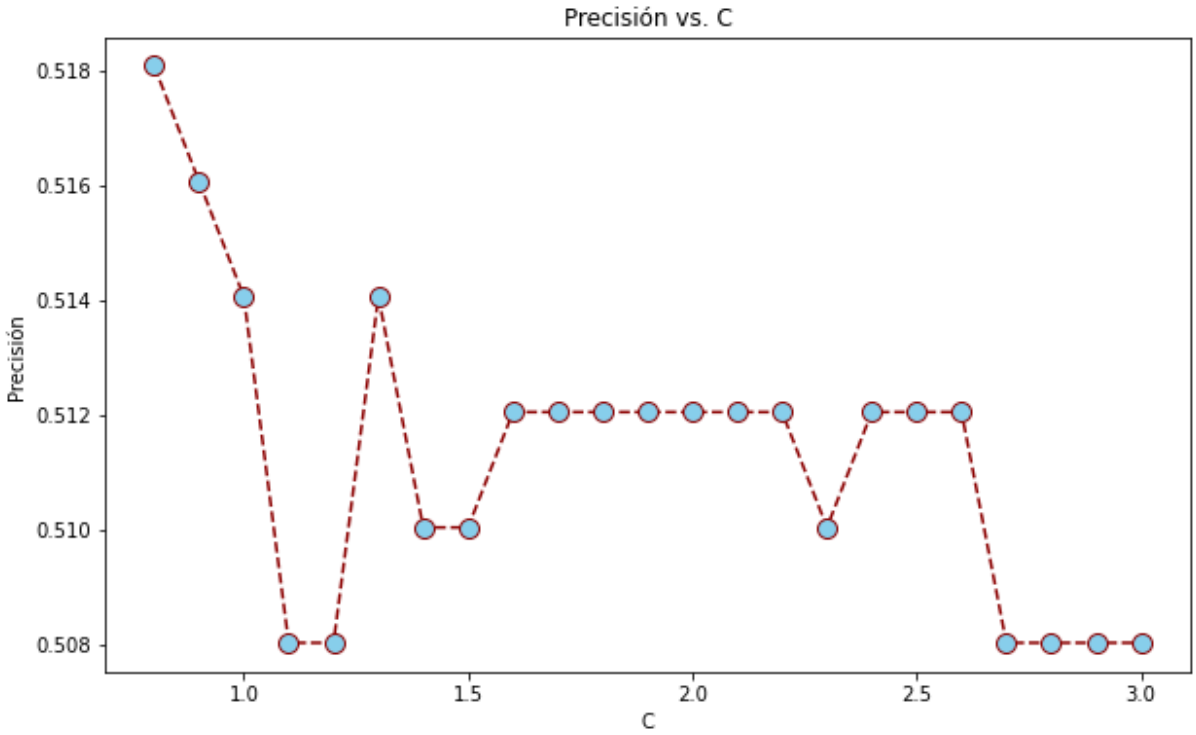

Figura 24. Precisión vs valor C para la máquina de soporte vectorial con kernel RBF

La precisión máxima para el modelo con kernel RBF fue de 51.81% con un parámetro de regularización de 0.8. Por esto, se obtuvo la matriz de confusión del modelo con estas características (Figura 25) y se muestra en la Figura 26.

```
#Se crea el modelo de kernel rbf y c=0.8
svm=SVC(C=0.8,kernel='rbf')
#Se entrena el modelo
svm.fit(X_train, y_train)
#Se grafica la matriz de confusión
plot_confusion_matrix(svm,X_test, y_test,labels=['Alto','Bajo'],cmap=plt.cm.Blues)
```

Figura 25.Código para generar la matriz de confusión de la máquina de soporte vectorial con kernel RBF

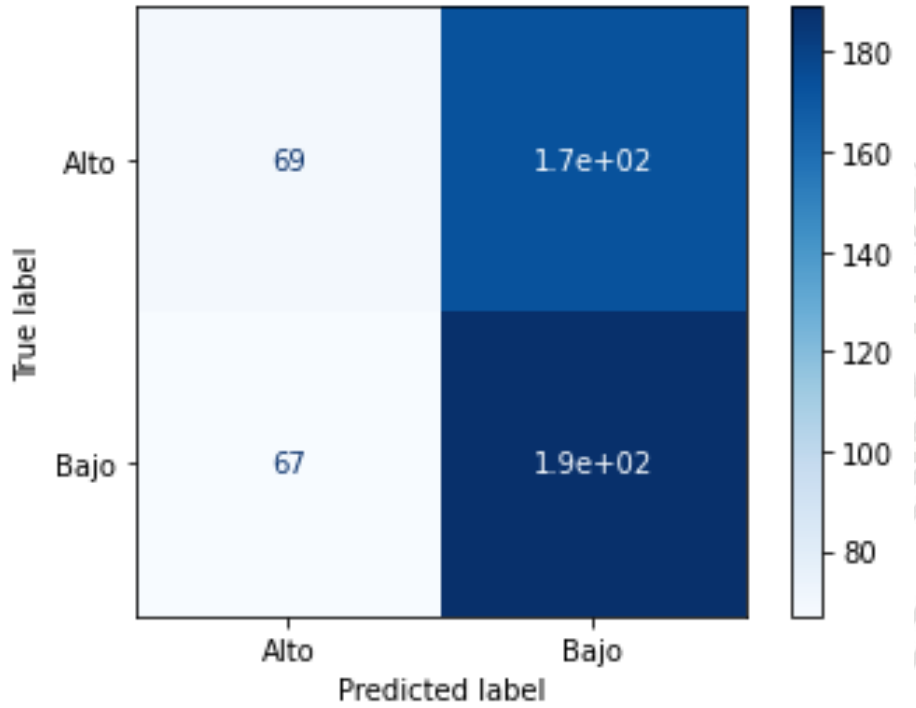

Figura 26. Matriz de confusión para la máquina de soporte vectorial con kernel RBF

Finalmente se definieron modelos con kernel polinomial de diferentes grados, entre los cuales, con el que se obtuvo una mayor precisión fue con el polinomio de grado 3. Por tanto, se obtuvieron los modelos con kernel polinomial grado 3 y con diferentes parámetros de regularización y el resultado se muestra en la Figura 27.

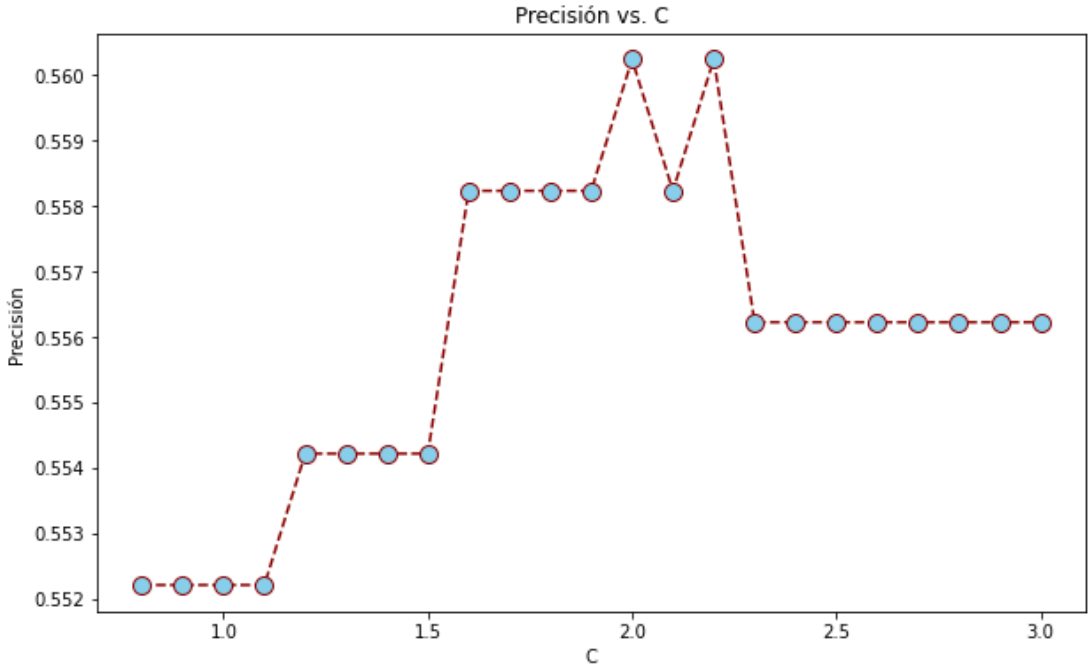

Figura 27. Precisión vs valor C para la máquina de soporte vectorial con kernel polinomial grado 3

Como se observa en la Figura 27 es posible determinar que la mayor precisión del modelo, que es de 56.02%, se encuentra con un parámetro de regularización C de 2. Por consiguiente, se obtuvo la matriz de confusión (Figura 28) del modelo con estas características y se muestra en la Figura 29.

```
#Se crea el modelo de kernel polinomial grado 3 y c=2
svm=SVC(C=2,kernel='poly',degree=3)
#Se entrena el modelo
svm.fit(X_train, y_train)
#Se grafica la matriz de confusión
plot_confusion_matrix(svm,X_test, y_test,labels=['Alto','Bajo'],cmap=plt.cm.Blues)
```

Figura 28.Código para obtener la matriz de confusión de la máquina de soporte vectorial con kernel polinomial grado 3

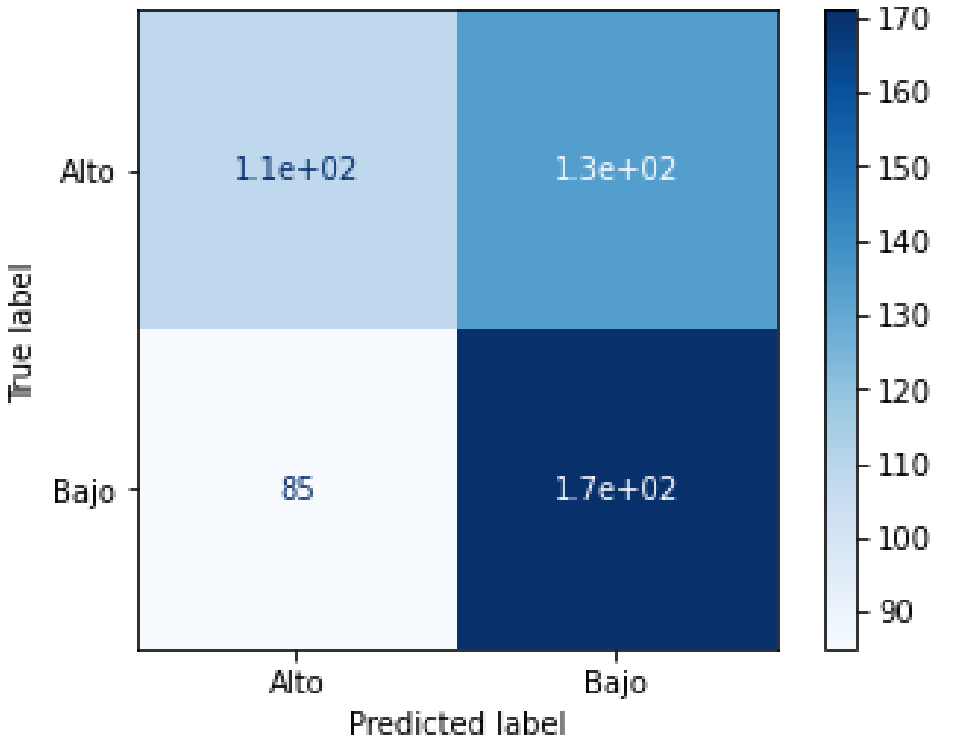

Figura 29. Matriz de confusión para la máquina de soporte vectorial con kernel polinomial grado 3

Por los resultados mostrados anteriormente es posible mencionar que, para la máquina de soporte vectorial, el modelo que mejor desempeño tuvo fue el modelo con kernel polinomial de grado 3 y con parámetro de regularización C de 2. Sin embargo, como se puede observar en la Figura 29, el modelo clasifica de manera adecuada la mayoría de los datos que se encuentran clasificados como riesgo bajo, pero clasifica de manera errónea la mayoría de los datos que se encuentran clasificados como riesgo alto.

Al analizar todos los modelos presentados anteriormente, es posible notar que los resultados no tuvieron una gran variación de una técnica a otra, teniendo como máxima precisión un 58.63% con el modelo de k vecinos más cercanos. Por esto, es importante mencionar que se debe trabajar bajo la base de datos, es decir, incluir una mayor cantidad de datos significativos para el etiquetado que se planteó en el presente proyecto. Además, en los modelos de Gradient Boosting y máquina de soporte vectorial, es posible tunear otros parámetros que permitan mejorar las predicciones con dichas técnicas.

6.5 Desarrollo de herramienta web

Se desarrolló una aplicación web (http://www.muertescovid.co/), en la cual se puede observar la información contenida en la base de datos creada en el presente proyecto. La aplicación fue desarrollada en Dash, el cuál es un framework de Python que permite el desarrollo de aplicaciones web para visualización de datos.

La aplicación contiene 3 pestañas, en la primera pestaña se puede observar la información por cada ciudad y se pueden seleccionar los meses que se quieren visualizar. Sin embargo, al abrir la página, se muestra la información de Colombia y de todos los meses contenidos en la base de datos. Esto se muestra en la Figura 30.

Figura 30. Aplicación web cuando es abierta en el navegador

En esta primera pestaña se pueden observar 4 gráficas diferentes. La primera de ellas es la distribución por sexo que se observa en la Figura 31.

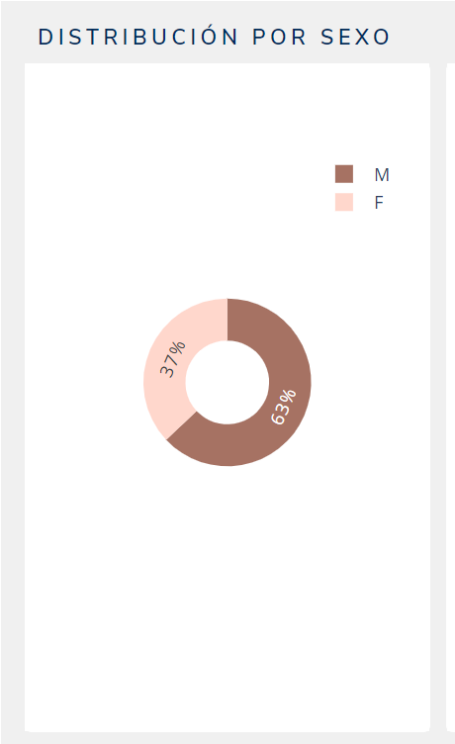

Figura 31. Gráfica de distribución por sexo de la aplicación web

En la segunda gráfica es posible observar la distribución por edad y esta se muestra en la Figura 32.

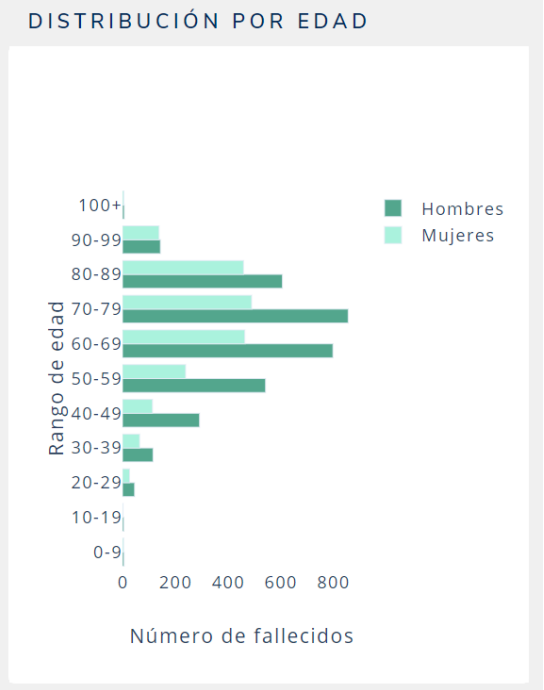

Figura 32. Gráfica de distribución por edad de la aplicación web

En la tercera gráfica se puede observar los días entre inicio de síntomas y muerte y esta se muestra en la Figura 33.

Figura 33. Gráfica de días entre inicio de síntomas y muerte de la aplicación web

Por último, en la primera pestaña, se muestra la gráfica del top de comorbilidades, incluyendo la opción de cambiar el rango de edad para observar las comorbilidades y esta gráfica se muestra en la Figura 34.

En la segunda pestaña se puede hacer una comparación del top de comorbilidades entre dos ciudades elegidas y esta gráfica se muestra en la Figura 35.

Por último, en la tercera pestaña se muestra el número de fallecidos por cada comorbilidad y es posible elegir entre las tres comorbilidades que se quieran observar. Esta gráfica se muestra en la Figura 36. La aplicación web obtenida es una herramienta útil para la visualización de la información relacionada con los fallecidos por COVID-19 en Colombia, teniendo en cuenta también las comorbilidades reportadas. Además, es interactiva y fácil de utilizar para que cualquier persona pueda visualizar la información que sea de su mayor interés.

Figura 34. Gráfica del top de comorbilidades de la aplicación web

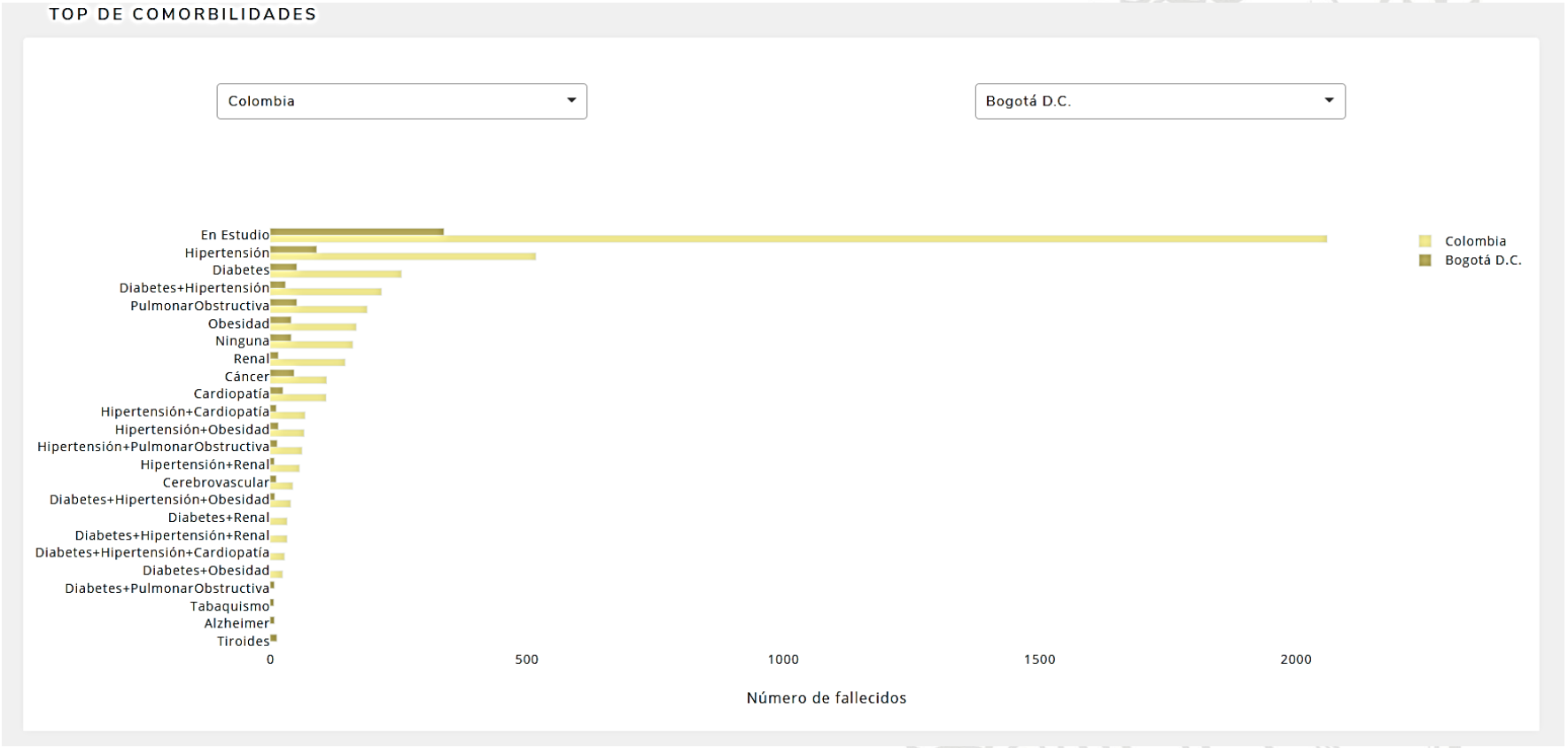

Figura 35. Gráfica de la comparación entre dos ciudades del top de comorbilidades de la aplicación web

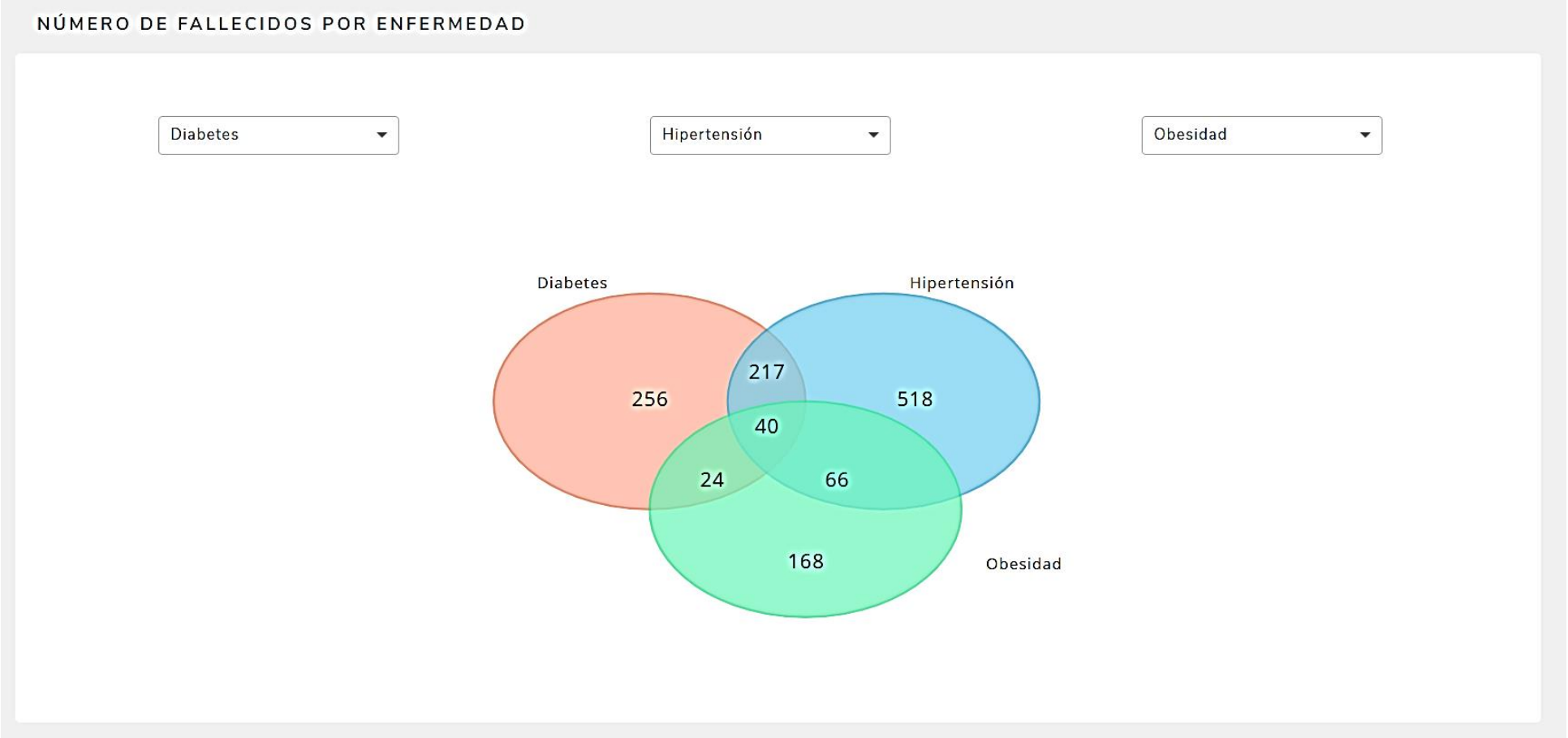

Figura 36. Gráfica de número de fallecidos por cada comorbilidad escogida en la aplicación web

7 Conclusiones

Las siguientes conclusiones se establecieron de acuerdo con los objetivos planteados.

- Los datos disponibles por el Ministerio de Salud y Protección Social de Colombia no fueron suficientes para una adecuada caracterización de la población de pacientes diagnosticados con SARS-CoV-2, debido a que no fue posible acceder a la información sobre las comorbilidades de todos los pacientes (vivos y fallecidos).
- Para la realidad actual de la pandemia, es de gran importancia contar con bases de datos que aporten información sobre las comorbilidades de los pacientes diagnosticados con SARS-CoV-2, debido a que esto sería de gran utilidad para predecir el riesgo que los pacientes tienen de presentar un cuadro clínico grave, según las comorbilidades que presente. Sin embargo, es necesario trabajar bajo la base de datos obtenida, incluyendo una mayor cantidad de datos significativos para la clasificación planteada.
- Para el análisis de las comorbilidades de los pacientes diagnosticados con SARS-CoV-2 es necesario implementar técnicas de aprendizaje automático supervisado y de clasificación, incluyendo desde las técnicas más simples como k vecinos más cercanos hasta técnicas más sofisticadas como las máquinas de soporte vectorial. Siempre y cuando sean implementados con las características adecuadas y con los suficientes datos significativos con respecto al etiquetado planteado.
- En el entrenamiento y validación de los modelos de Machine Learning es importante tunear todos los hiperparámetros que se consideren importantes para cada modelo. Por tanto, es de gran relevancia mencionar que las predicciones de las técnicas obtenidas pueden mejorar significativamente si se varían algunos de los parámetros de los modelos.
- La disponibilidad de una herramienta web interactiva y fácil de utilizar para visualizar los datos de las personas fallecidas a causa del COVID-19 en Colombia, que incluya, además, la información sobre las comorbilidades asociadas a dichos casos es de gran utilidad, debido a que esta información no se tiene disponible en ninguna otra plataforma para ser visitada por el público en general.

8 Referencias bibliográficas

[1] "Coronavirus disease (COVID-19) pandemic." [Online]. Available: https://www.who.int/emergencies/diseases/novel-coronavirus-2019.
[2] "Q&A on coronaviruses (COVID-19)." [Online]. Available: https://www.who.int/es/emergencies/diseases/novel-coronavirus-2019/advice-for-public/q-a-coronaviruses.
[3] "Boletines casos COVID-19 Colombia," 2020. [Online]. Available: https://www.ins.gov.co/Paginas/Boletines-casos-COVID-19-Colombia.aspx#InplviewHash5872a312-02d0-4090-aa8a-7716dd9fc4df=Paged%3DTRUE-p_SortBehavior%3D0-p_FileLeafRef%3D2020%252d08%252d10%252exlsx-p_ID%3D186-PageFirstRow%3D151.
[4] "Semana." [Online]. Available: https://www.semana.com/.
[5] "La OMS publica directrices para ayudar a los países a mantener los servicios sanitarios esenciales durante la pandemia de COVID-19." [Online]. Available: https://www.who.int/es/news-room/detail/30-03-2020-who-releases-guidelines-to-help-countries-maintain-essential-health-services-during-the-covid-19-pandemic.
[6] Elsevier Connect, "¿Qué es el 'machine learning'?," 2018. [Online]. Available: https://www.elsevier.com/es-es/connect/ehealth/que-es-el-machine-learning-salud-digital.
[7] D. Patrik, F. James, and S. Basu, "Machine Learning for Health Services Researchers," *Elsevier*, 2019.
[8] M. M. Kowalik, P. Trzonkowski, M. Łasińska-Kowara, A. Mital, T. Smiatacz, and M. Jaguszewski, "COVID-19 — Toward a comprehensive understanding of the disease," *Cardiol. J.*, vol. 27, no. 2, pp. 99–114, 2020, doi: 10.5603/CJ.a2020.0065.
[9] K. A. Walsh *et al.*, "SARS-CoV-2 detection, viral load and infectivity over the course of an infection," *J. Infect.*, vol. 81, no. 3, pp. 357–371, 2020, doi: 10.1016/j.jinf.2020.06.067.
[10] Y. Liu *et al.*, "Viral dynamics in mild and severe cases of COVID-19," *Lancet Infect. Dis.*, vol. 20, no. 6, pp. 656–657, 2020, doi: 10.1016/S1473-3099(20)30232-2.
[11] "La comorbilidad." [Online]. Available: https://www.drugabuse.gov/es/informacion-sobre-drogas/la-comorbilidad.
[12] M. Kumar, K. Kuroda, and K. Dhangar, "Prevalence of comorbidities among individuals with COVID-19: A rapid review of current literature," no. January, 2020.
[13] M. learning Blog, "What is Machine Learning? A definition," 2017. [Online]. Available: https://expertsystem.com/machine-learning-

definition/.
[14] T. M. D. Ebbels, "Chapter 7 - Non-linear Methods for the Analysis of Metabolic Profiles," in *The Handbook of Metabonomics and Metabolomics*, J. C. Lindon, J. K. Nicholson, and E. Holmes, Eds. Amsterdam: Elsevier Science B.V., 2007, pp. 201–226.
[15] "Naive Bayes." [Online]. Available: https://scikit-learn.org/stable/modules/naive_bayes.html.
[16] A. A. S. Castro, "Una introducción a los Árboles de Decisión," 2020. [Online]. Available: https://www.grupodabia.com/post/2020-05-19-arbol-de-decision/.
[17] A. Callens, D. Morichon, S. Abadie, M. Delpey, and B. Liquet, "Using Random forest and Gradient boosting trees to improve wave forecast at a specific location," *Appl. Ocean Res.*, vol. 104, no. March, p. 102339, 2020, doi: 10.1016/j.apor.2020.102339.
[18] N. L. Costa, L. A. G. Llobodanin, I. A. Castro, and R. Barbosa, "Using Support Vector Machines and neural networks to classify Merlot wines from South America," *Inf. Process. Agric.*, vol. 6, no. 2, pp. 265–278, 2019, doi: 10.1016/j.inpa.2018.10.003.
[19] M. Colleen, "Learning Algorithm," 2015. [Online]. Available: https://www.sciencedirect.com/topics/computer-science/learning-algorithm.
[20] M. Rodriguez *et al.*, "Clustering algorithms: A comparative approach," 2019, doi: 10.1371/journal.pone.0210236.
[21] "Selección del número óptimo de Clusters," 2016. [Online]. Available: https://jarroba.com/seleccion-del-numero-optimo-clusters/.
[22] M. Vlachos, "Dimensionality Reduction," in *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb, Eds. Boston, MA: Springer US, 2010, pp. 274–279.
[23] X. Zhu, "Semi-Supervised Learning," in *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb, Eds. Boston, MA: Springer US, 2010, pp. 892–897.