BUAP

BENÉMERITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

Algoritmos de compresión de datos aplicados a historiales clínicos para el apoyo en el diagnóstico de Diabetes Mellitus tipo II

Tesis presentada por:

Ing. Juan Manuel Cancino Gordillo

Para obtener el grado de:

Maestria en ciencias de la computación

Dirigida por:

Dr. David Eduardo Pinto Avendaño

Dra. Mireya Tovar Vidal

Puebla, México
Julio 2021

Dedicatoria

Para mis padres, quienes han sido mi inspiración y luz, quienes me enseñaron a volar sin temor a caer y me han dado todo el amor de este mundo.

A mi hermano mayor quien me ha apoyado, orientado y guiado pero sobre todo ha sido mi mejor amigo.

Agradecimientos

En primer lugar me gustaría agradecer a mis asesores, la Dra. Mireya Tovar Vidal y el Dr. David Eduardo Pinto Avendaño, por la ayuda incondicional, enseñanzas académicas y paciencia que han tenido conmigo para cumplir con este nuevo logro.

Extiendo mis mas sinceros agradecimientos a la Facultad de ciencias de la Computación, quienes nos ha apoyado con los materiales, el espacio y el tiempo para el desarrollo del proyecto. También agradezco al Consejo Nacional de Ciencia y Tecnología (CONACyT) por otorgar la beca que permitió realizar mis estudios (CVU #972078). A mis compañeros de trabajo presentes en el laboratorio, los cuales a pesar de estar a distancia o en diferentes lineas de investigación, siempre mostraron un apoyo incondicional y estuvieron presentes.

De igual manera al Dr. Neri Salvador Cancino Hernández, quien ha aportado ciertos registros médicos utilizados en este trabajo, la ayuda y opinión no solo como doctor en salud, si no como un amigo, orientándome y ayudándome en la realización de este trabajo.

Finalmente quiero agradecer a toda mi familia por su confianza, preocupación y especialmente su apoyo, ya que sin ellos no sería la persona que ha llegado hasta el día de hoy.

Resumen

Una de las enfermedades más importantes a nivel mundial en salud pública es la Diabetes Mellitus (DM), ya que esta es una de las enfermedades no transmisibles más severa, frecuente y con diversas complicaciones crónicas. Existen dos variantes de la DM, tipo I y tipo II. En este documento nos enfocamos en la detección de Diabetes Mellitus tipo II (DMT2), en donde el cuerpo no procesa de manera correcta la glucosa en la sangre dejando mucho de este material circulando dentro del sistema sanguíneo. En este documento proponemos un método para la detección de factores de riesgo en pacientes que padecen la enfermedad conocida como DMT2 con distintos conjuntos de datos estructurados aplicando algoritmos de clasificación junto a algoritmos de reducción de términos.

En este documento abarcaremos los antecedentes que dieron origen al proyecto de investigación junto con los objetivos. Nuestra propuesta es usar un análisis de relación entre atributos junto con algoritmos de reducción de términos (PCA y NMF) para reducir lo más posible los conjuntos de datos sin realizar una clasificación pobre de personas con la enfermedad DMT2. El conjunto de datos reducido que mejor se comporte con las métricas de evaluación de minería de datos (precisión, exhaustividad, exactitud y F1) es considerado la lista de factores de riesgo del conjunto de datos. En donde los resultados obtenidos del algoritmo *Random forest* obtiene una exactitud del 98.7 % utilizando un conjunto de datos provenientes de China con nueve atributos y 94.2 % de exactitud con un conjunto de datos proveniente de México utilizando seis atributos. Mientras que un conjunto de datos de India supera las métricas del estado del arte utilizando los algoritmos *TreeJ48*, *Naïve Bayes*, *Random forest* y SMO. Obteniendo resultados de 86 %, 77.0 % y 78.5 % respectivamente con seis atributos.

Abstract

One of the most important diseases worldwide in public health is Diabetes Mellitus (DM), since this is one of the most severe and frequent non-communicable diseases with various chronic complications. There are two variants of DM, type I and type II. In this document we focus on the detection of Type II Diabetes Mellitus (T2DM), where the body does not correctly process glucose in the blood, leaving much of this material circulating within the blood system. In this document we propose a method for the detection of risk factors in patients suffering from the disease known as T2DM with different structured data sets applying classification algorithms together with term reduction algorithms.

In this document we will cover the background that gave rise to the research project together with the objectives. Our proposal is to use an attribute relationship analysis together with term reduction algorithms (PCA and NMF) to reduce the data sets as much as possible without performing a poor classification of people with T2DM. The reduced data set that best performs with the data mining evaluation metrics (precision, recall, accuracy, and F1) is considered the list of risk factors in the data set. Where the results obtained from the *Random forest* algorithm obtain an accuracy of 98.7 % using a data set from China with nine attributes and 94.2 % accuracy with a data set from México using six attributes. While a dataset from India outperforms state-of-the-art metrics using *TreeJ48*, *Naïve Bayes*, *Random forest* and SMO algorithms. Obtaining results of 86 %, 77.0 % and 78.5 % respectively with six attributes.

Índice general

Dedicatoria i

Agradecimientos ii

Resumen iii

Abstract iv

Índice de figuras vii

Índice de tablas ix

1. Introducción **1**

1.1. Antecedentes 2

1.2. Objetivos 3

1.3. Metodología 3

2. Estado del arte **5**

3. Marco teórico **11**

3.1. Análisis de datos 11

3.1.1. Escala de características 12
3.2. Inteligencia artificial 14
3.3. Aprendizaje automático 15
3.4. Clasificadores 18
3.4.1. Árboles de decisión 18
3.4.2. Naïve Bayes 21
3.4.3. Random forest 24
3.4.4. Máquinas de soporte vectorial 25
3.5. Big data 27
3.5.1. Análisis de componentes principales 28
3.6. Procesamiento de lenguaje natural 32
3.6.1. Recuperación de información 33
3.7. Paquetes esenciales de Python 35
3.7.1. Pandas 35
3.7.2. PyDoxc 35
3.7.3. BeautifulSoup 35
3.7.4. Scikit-learn 36

4. Propuesta de solución **37**

4.1. Recopilación y análisis de datos 37
4.2. Pre-Procesado 39
4.3. Clasificación 42
4.4. Evaluación de algoritmos 42

5. Resultados **45**

5.1. Conjuntos de datos 45

5.1.1. Primer conjunto de datos (PIDD) 45

5.1.2. Segundo conjunto de datos (China) 46

5.1.3. Tercer conjunto de datos (México) 46

5.2. Resultados del pre-procesado 48

5.2.1. Primer conjunto de datos (PIID) 48

5.2.2. Segundo conjunto de datos (China) 51

5.2.3. Tercer conjunto de datos (México) 54

5.3. Clasificación y evaluación 56

5.3.1. Primer conjunto de datos (PIDD 57

5.3.2. Segundo conjunto de datos (China) 58

5.3.3. Tercer conjunto de datos (México) 60

6. Conclusiones **63**

Bibliografía 65

Anexos **70**

6.1. Anexo B 73

Índice de figuras

3.1. Min-Max scaling [22]. 13
3.2. Normalización de datos [22]. 13
3.3. Ejemplo básico de clasificación binaria [25]. 17
3.4. Estructura de un árbol de decisión [26]. 19
3.5. Ejecución de un árbol de decisión [26]. 20
3.6. Hiper-plano de separación en dos dimensiones [27]. 25
3.7. Tres posibles hiper-planos de un caso perfectamente separable [27]. . . . 26
3.8. Hiper-plano óptimo de separación [27]. 26
3.9. Escenarios de tipo no lineal [27]. 27
3.10. Pétalo (petal) y sépalo (sepal) de una flor [31]. 29
3.11. Auto-vectores y auto-valores. 31
3.12. Varianza explicada. 31
3.13. Representación gráfica del conjunto de datos en dos dimensiones. 32

4.1. Diagrama de flujo. 38
4.2. Extracción de datos y recuperación de información. 39
4.3. Procedimiento para sustitución por media. 40

4.4. Representación de matriz de confusión. 42

5.1. Matriz de correlación del conjunto de datos PIDD. 49

5.2. Correlación glucosa - insulina. 49

5.3. Correlación grosor de piel – IMC. 50

5.4. Representación gráfica de auto-valores con conjunto de datos PIDD. . . 50

5.5. Matriz de correlación del conjunto de datos China. 52

5.6. Relación colesterol total – colesterol *LDL*. 52

5.7. Representación gráfica de auto-valores con conjunto de datos China. . . 53

5.8. Matriz de correlación del conjunto de datos México. 55

5.9. Representación de auto-valores con conjunto de datos México. 55

6.1. Pagina principal del sistema Docx. 71

6.2. Consultas de un paciente. 71

6.3. Reporte generado automáticamente. 72

6.4. Interfaz del programa. 73

6.5. Texto mostrado en consola. 74

6.6. Concentrado de datos. 75

Índice de tablas

3.1. Representación del conjunto de datos [31]. 30

4.1. Estructura del conjunto de datos. Los puntos en la tabla representan más datos. 41

5.1. Descripción del conjunto de datos PIDD [36] 46

5.2. Descripción del conjunto de datos China [35]. 47

5.3. Descripción del conjunto de datos México. 47

5.4. Resultados de NMF con conjunto de datos PIDD. 51

5.5. Conjuntos de datos generados por los algoritmos. 51

5.6. Resultados de NMF con conjunto de datos China. 53

5.7. Conjuntos de datos generados por los algoritmos. 54

5.8. Resultados de NMF con conjunto de datos México. 56

5.9. Conjuntos de datos resultantes de la sección. 56

5.10. Resultados del conjunto de datos PIDD usando *TreeJ48*. 57

5.11. Resultados del conjunto de datos PIDD usando *Naïve Bayes*. 58

5.12. Resultados del conjunto de datos PIDD usando SMO. 58

5.13. Resultados del conjunto de datos China usando *TreeJ48*. 59

5.14. Resultados del conjunto de datos de China usando *Naïve Bayes*. 59

5.15. Resultados del conjunto de datos China usando *Random Forest*. 60

5.16. Matriz de confusión resultante con atributos completos. 60

5.17. Matriz de confusión resultante con pre-procesado. 60

5.18. Resultados del conjunto de datos México usando *TreeJ48*. 61

5.19. Resultados del conjunto de datos México usando *Naïve Bayes*. 61

5.20. Resultados del conjunto de datos México usando *Random Forest*. . . . 62

5.21. Mejor resultado por cada conjunto de datos. 62

Capítulo 1

Introducción

La DM es una enfermedad severa a nivel mundial, ya que conlleva diversas complicaciones crónicas y la atención de varios especialistas para tratar la enfermedad de una persona. En el campo médico el diagnóstico es la parte más importante a la hora de tratar a una persona, ya que el médico utiliza sus conocimientos para detectar patrones en el comportamiento o estudios médicos de un paciente.

Existen dos variantes de la DM, tipo I y tipo II, donde la DMT1 es conocida como una afección crónica donde el páncreas produce poco o nada de insulina, mientras que la DMT2 es conocida por afectar a la manera en la que el cuerpo procesa el azúcar en sangre (glucosa). Para detectar a una persona con esta enfermedad se realizan encuestas conocidas como factores de riesgo, las cuales son una serie de preguntas sobre sus actividades diarias y ciertas medidas antropométricas, basándose en esta encuesta un experto puede diagnosticar a una persona dado a los resultados de esta y si se encuentra en riesgo de desarrollar la enfermedad de diabetes (pre-diabético).

El propósito del trabajo es realizar un análisis en los parámetros que utilizan los médicos para detectar la DMT2, realizar una limpieza de datos, compresión de datos y construir un modelo matemático donde se utilicen la menor cantidad de atributos. Ya que estos se traducen a la realización de un estudio para determinar si una persona padece la enfermedad o no. Posteriormente crear un modelo de clasificación eficiente basándonos en los atributos obtenidos del análisis para encontrar los factores de riesgo más representativos de la DMT2.

1.1. Antecedentes

Las enfermedades crónicas no transmisibles (ECNT) son enfermedades que constituyen un serio problema de salud por su prevalencia y mortalidad, además de ser uno de los casos más comunes, complejos y costosos en casos clínicos, dado a que requieren de asistencia simultanea de diferentes especialistas [1]. Estas enfermedades han representado la causa principal de mortalidad en la mayoría de los países, con un estimado mundial de 63 % en el 2015 [2]. En México la enfermedad conocida como Diabetes Mellitus tipo 2 (MDT2) ha presentado una de las causas principales por muerte desde el año 2005 con una prevalencia del 11.8 % de muertes y con costos médicos asociados aproximados a 450 millones de dólares [3].

Para la atención de una amplia cantidad de pacientes se han desarrollado distintos mecanismos que puedan personalizar y describir la condición del conocimiento de los pacientes (historial clínico o expedientes clínicos) para combatir una enfermedad. El principal problema es la conversión de información existente en un conocimiento que pueda ser operativo y funcional en el contexto de su aplicación. El ejemplo perfecto es el expediente clínico electrónico (ECE), que desde el 2011 se impuso como un estándar para normalizar las funcionalidades, datos e historia clínica de un paciente garantizando confidencialidad de la identidad de los pacientes [4]. La cuál nos daría la posibilidad de tomar ciertos dados conocidos como factores de riesgo y predecir resultados a las ECNT.

La historia clínica es uno de los elementos esenciales del sistema de información asistencial enfocado en el paciente [5] dado a que en el ámbito nacional, la transformación electrónica de la historia clínica se está llevando a cabo para la integración de varios subsistemas parciales de información. Des-afortunadamente el ECE padece los mismos males del Sistema Nacional de Salud, dado que opera de manera fragmentada, desarticulada y con baja difusión. El mayor problema es que las instituciones de salud operan con una versión propia del ECE, mientras que en el sector privado operan con plataformas cibernéticas desvinculadas del sector público. Desde la reforma constitucional del 11 de junio de 2013 es obligatorio usar un ECE en todo el sector de salud, sin embargo, no ha existido la voluntad política para instrumentarlo [6].

1.2. Objetivos

El objetivo general y los objetivos específicos propuestos para esta investigación son:

- **Objetivo general:**
  - Implementar métodos de compresión de datos en historiales clínicos para clasificar de manera correcta la enfermedad Diabetes Mellitus tipo II.
- **Objetivos específicos:**
  - Obtener expedientes médicos, historiales clínicos o bases de datos para la recopilación de datos.
  - Realizar un módulo para la extraer atributos en datos no estructurados relacionados a un historial clínico y convertirlos datos estructurados.
  - Desarrollar un módulo para el pre-procesamiento de los datos estructurados del historial clínico.
  - Desarrollar e implementar algoritmos de compresión de atributos con los conjuntos de datos obtenidos.
  - Implementar modelos de aprendizaje automático utilizando los datos obtenidos de la historia clínica.

1.3. Metodología

La metodología integra cuatro etapas esenciales para el desarrollo de la investigación. Las cuales consisten en:

- **Recopilación y análisis de datos:** Donde se da una breve explicación de la estructura y contenido de los conjuntos de datos utilizados en la investigación. Tiene como objetivo buscar los factores de riesgo de DMT2 dentro de los conjuntos de datos, la cual estará comprimida en una estructura ordenada proveniente de la historia clínica, utilizando algoritmos de compresión de datos. El procedimiento básico consiste en cinco fases: especificación de los requisitos, recopilación de los datos, pre-procesamiento de datos, limpieza de datos y análisis de los datos. En el trabajo las dos primeras partes son omitidas, ya que se utilizan conjunto de

datos con una orientación y se comienzan a trabajar desde el pre-procesamiento de los datos y culminar en el análisis de estos mediante algoritmos compresión de datos para determinar que atributos son más relevantes dentro del conjunto de un conjunto de datos.

- **Pre-procesamiento de datos:** Donde se explica la metodología usada en la investigación para tratar cada conjunto de datos para la eliminación de datos que faltan, no válidos o para evitar una pobre clasificación, ya que ciertos algoritmos empiezan a clasificar erróneamente al tener una gran cantidad de atributos no relevantes dentro de la información recopilada.
- **Selección de algoritmos / clasificación:**Donde se realizará una implementación de los algoritmos más utilizados dentro de la rama de clasificadores y aprendizaje automático mediante herramientas o códigos propios para descubrir que atributos son los encargados de realizar la clasificación. Con el propósito de comparar esta clasificación con una realizada mediante algoritmos de compresión de datos y determinar una lista de atributos principales que puedan ser considerados factores de riesgo.
- **Evaluación de los algoritmos:**Donde se explican las métricas y métodos que se utilizan para llevar acabo la evaluación de un algoritmo de aprendizaje automático.

El documento de tesis está dividido en seis capítulos. En el capítulo 2 se presentan los trabajos relacionados al tema de clasificadores utilizando diferentes algoritmos y métodos para limpieza de archivos, junto a los porcentajes alcanzados por cada trabajo. En el capítulo 3 se da una breve explicación a los términos que fueron utilizados en el desarrollo del trabajo. En el capítulo 4 se da una breve explicación a las metodologías, algoritmos y métricas de evaluación utilizadas dentro del trabajo. En el capítulo 5 se presentan los resultados aplicando la metodología explicada para continuar con las conclusiones del trabajo en el capítulo 6, para finalizar con la bibliografía consultada y anexos del trabajo.

Capítulo 2

Estado del arte

La guía de actualización en diabetes realizada en España en el año 2015 [7] nos muestra el estudio donde utiliza un conjunto de atributos utilizados de la *National Institute for Health and Care Excellence* (NICE), para mostrarnos un listado con razones de los factores de riesgo no modificables y modificables relacionados a la DMT2. Determinando los siguientes atributos como factores de riesgo, la edad, IMC, alteración de glucosa y sedentarismo.

En el siguiente año Llorante *et al.* [8] presentaron un análisis de los factores de riesgos asociados con la aparición de diabetes mellitus tipo 2. El análisis se realizó con un conjunto de datos compuesto de 20,396 personas registradas entre el año 2011 y 2012, utilizando como comparativa las medidas de varianza o la prueba U de *Mann-Whitney*, calculando *Odds-ratio* e intervalos de confianza del 95 % en la regresión lógica. En el análisis de los datos llegan a conclusión los pacientes diagnosticados con diabetes presentan valores altos en los atributos de presión arterial, índice de colesterol LDL/HDL, edad e índice de masa corporal, teniendo factores de riesgo independientes como, antecedentes familiares de diabetes y síndrome metabólico.

Durante el 2015 se intentó detectar mediante el algoritmo *Naïve Bayes* enfermedades relacionadas con el corazón [9], con el objetivo de reducir la cantidad de pruebas extras para la detección de enfermedades cardiovasculares, en el artículo nos presenta trabajos relacionados donde se ve el método de clasificación utilizado y la cantidad de atributos usados dentro del conjunto de datos. En el caso del trabajo presentado por los autores K. Vembandasamy *et al.* cuentan con un conjunto de datos compuesto de 500 instancias y atributos de tipo discreto (para normalizar ciertas partes de los datos) junto con

atributos continuos. Utilizando la herramienta *WEKA* y una separación de archivos 70/30 obtiene clasificaciones correctas de 86.4198 % utilizando un algoritmo simple de *Naïve Bayes*, mientras que un modelo modificado por ellos llega a clasificar un 74 % correctamente, con 71 %, 74 % y 71.2 % en las métricas de precisión, exhaustividad y F_1 respectivamente, superando los porcentajes presentados en el trabajo relacionado.

Orlando A. Chan *et al.* [10] presento una investigación sobre un conjunto de datos de 768 pacientes, donde los registros están basados en mujeres para la detección de diabetes gestacional con la intención de crear posteriormente un sistema experto que apoye a los diagnósticos de diabetes. Algunas de sus variables consideradas en su estudio son: glucosa, presión sanguínea, insulina y edad. Con el conjunto de datos los autores determinan que el atributo glucosa sobresale de manera importante para determinar si una persona padece de diabetes, pero, estas variables no siempre generan resultados confiables y requiere de seguimiento un seguimiento para determinar la enfermedad. Los autores utilizaron herramientas como *WEKA* y *BigML* para la generación de sus modelos consiste en un grafo con estructura de árbol para la clasificación (árboles de decisión J48) donde cada nodo representa una pregunta y cada rama corresponde a una respuesta concreta de la pregunta. La conclusión de los autores muestra los modelos generados por la herramienta *WEKA* obtenían una precisión del 70 % con los pacientes que no tienen la enfermedad y un 63 % de precisión positiva con el conjunto de datos recopilado de 768 pacientes.

En el trabajo presentado por AlJarullah *et al.* [11] nuevamente se hace la mención de árboles decisión para el descubrimiento de personas con DMT2 enfocados en un conjunto de datos de mujeres utilizado frecuentemente para la detección de la diabetes. Aquí se distribuyen en dos etapas, la primera parte es el pre-procesamiento de datos, el cual intenta mejorar los datos aplicando métodos eliminación de registros, junto con la discretización de datos para manejar la información ausente, estos métodos son utilizados mayormente en proyectos orientados a minería de datos ya que ayuda a eliminar instancias del conjunto de datos. Mientras que la segunda fase consiste en la construcción del modelo predictivo con la herramienta *WEKA*. Los autores llegan a la conclusión que el pre-procesado y discretización de los datos se alcanza un modelo con un 78.1768 % en la métrica de precisión. Demostrando que realizar un pre-procesado de datos mejora la clasificación de instancias del conjunto de datos utilizado.

El trabajo de P. Hemant *et al.* [12] propone una serie de algoritmos utilizados en la rama de minería de datos como: *Sequential Minimal Optimization* (SMO), *random forest*, *Naïve Bayes*, entre otros para comparar el rendimiento de los algoritmos de clasificación y determinar que algoritmos posee una mayor exactitud al realizar la clasificación de un conjunto de datos. Los autores siguen los lineamientos de limpiar los

datos no requeridos para el estudio, como interpretar los datos que faltan del conjunto de datos, el cuál no está compuesto de un sólo lugar según la descripción que nos proporcionan. Para la evaluación de los algoritmos utilizan el método conocido como *Cross-validation* junto a las métricas precisión, exactitud y F_1, en donde dividen su conjunto de datos con una relación 50:50. Los autores mencionan que la relación que utiliza para su conjunto de datos no es ideal, ya que para este tipo de evaluaciones es mejor seccionar en tres partes el conjunto de datos. En los resultados y conclusiones del trabajo nos hace la mención de los resultados de la precisión del algoritmo J48, llegando en el mejor de los casos al 73.82 % de exactitud al clasificar.

En el trabajo de Vijayan *et al.* [13] nos explica la efectividad de los sistemas de soporte de decisiones junto alguno de los factores o atributos que se toman en cuenta para la detección de la enfermedad conocida como DM. Ya que dado estos atributos es capaz de aplicar algoritmos de aprendizaje automático para hacer una clasificación de los registros. En su trabajo nos presenta cuatro algoritmos diferentes aplicados en un conjunto de datos de mujeres de 768 registros con la información de: cantidad de embarazos, glucosa, presión sanguínea, índice de masa corporal, entre otros. Simplemente nos presenta un resumen de los datos y no la fuente de estos mismos. Como todo algoritmo de aprendizaje automático pasa por la limpieza de datos y manejo de datos que faltan para llevar acabo los algoritmos de clasificación como: árboles de decisión, SVM (*Support Vector Machine*), *Naïve Bayes* y *Desicion stump*. Al final de su trabajo nos muestra la precisión de cada uno de estos algoritmos los cuales llegan a porcentajes de 76 %, 79.68 %, 78.1 %, 74.47 % respectivamente. Como notas finales a estos porcentajes los autores aclaman que estos porcentajes pueden ser mejorados si se implementan diferentes clasificadores como redes neuronales (ANN) o *K-Nearest Neighbor*.

En el 2015 Sadri Sa'di *et al.* [14][14] presento una comparación entre algoritmos de minería de datos para la detección de diabetes utilizando la base de datos *Pima Indians Diabetes Dataset*, el cual consiste de 768 registros de mujeres con nueve atributos diferentes como: número de embarazos, glucosa, presión arterial, grosor de la piel, IMC, entre otros. En el trabajo nos presenta tres algoritmos de minería de datos como árboles de decisión J48, *Naïve Bayes* y *Radial Basis Function Network* (RBF). Nos da una breve introducción a cada uno de estos algoritmos para mostrar terminar con una tabla comparativa con las métricas de comparación como Presicion, Recall, y F_1. Para demostrar que el algoritmo *Naïve Bayes* alcanzó la precisión más alta con un 76.95 % para diagnosticar DMT2. En este trabajo no muestra nada del pre-procesamiento del conjunto de datos como los otros trabajos que se han presentado hasta ahora, en caso contrario en las conclusiones remarca que 230 registros fueron utilizados para las pruebas, esto puede verse reflejado en las métricas que presenta, un poco bajo en comparación a los trabajos reportados utilizando el mismo método.

Sajida P. *et al.* [15] presentó en el 2016 una metodología experimental para la detección de DM basada en los árboles de decisión J48 combinado con el método adaboost, es una buena alternativa para clasificar a los pacientes diabéticos. Su trabajo está completamente desarrollado en la herramienta utilizada en minería de datos conocida como *WEKA*. En este trabajo se utiliza un conjunto de datos obtenidos de la base de datos *CPCSSN* que contiene información demográfica, mediciones clínicas y valores de laboratorio del 2003 al 2013, donde se llegan a los 667,907 registros de los cuales 40,042 son pacientes diabéticos, aproximadamente el 6 % de pacientes. Menciona y desglosa los parámetros de los algunos factores de riesgo que se toman en cuenta a la hora de la detección de diabetes en una persona como: el colesterol HDL, triglicéridos, IMC y glucosa. Como fue mencionado, el trabajo muestra la clasificación del conjunto de datos, pero no utiliza ninguna métrica para aclamar que la clasificación que está realizando es completamente válido. Pero se puede ver en su trabajo que la metodología experimental supera la clasificación realizada por el algoritmo J48.

En el año reciente H. Deberneh *et al.* [16] presentó un modelo de aprendizaje automático para la detección de diabetes tipo II del año siguiente utilizando registros del año anterior en una población coreana. El conjunto de datos utilizados proviene de un instituto privado basado en expedientes electrónicos del 2013 al 2018. Para la construcción del modelo se utilizaron métodos como *ANOVA* (analisis of variance), chi-cuadrada y recursividad para la eliminación de atributos no relevantes. Para la clasificación, los autores presentaron varios algoritmos, entre ellos SVM, *random forest* y regresión logística, con el objetivo de clasificar tres instancias (no diabético, pre-diabético y diabético). Los autores al realizar la eliminación determinaron que los atributos de historial familiar, fumar, tomar y actividad física dieron los resultados más bajos para la selección de factores relevantes. Para la evaluación los autores separaron el conjunto de datos en dos (*test* y *training*) para que mediante una validación de *k-folds* utilizara el conjunto de entrenamiento y luego utilizar el conjunto de datos de prueba para evaluar el rendimiento del algoritmo. De acuerdo con los resultados, el desempeño de los algoritmos varia del 71 % al 73 % en la precisión de los modelos generados, siendo SMV el algoritmo con mejor rendimiento en comparación con regresión linear.

En temas recientes, la pandemia relacionada a COVID-19 ha dado paso a analistas de datos a buscar características de personas que padecen dicha enfermedad con complicaciones en la población mexicana [17]. Donde con un conjunto de datos del instituto epidemiológico mexicano compuesto de 23,593 pacientes donde 3,844 padecían de la enfermedad conocida como Coronavirus y 1,306 fueron diagnosticadas con otras enfermedades respiratorias. Para el desarrollo del análisis utilizaron modelos de regresión logística multi-variable para explorar los factores asociados con los casos severos de COVID-19 en pacientes ingresados. En las conclusiones del trabajo llegan a los re-

sultados que los parámetros de diabetes, obesidad e hipertensión son significativamente asociados con los pacientes reportados de gravedad con la enfermedad COVID-19, remarcando que la asociación de la obesidad ganaba peso conforme la edad era mayor.

De una manera similar al estudio realizado en México, Bertrand *et al.* [18] presenta un estudio realizado en centros de Francia especializado a las características del fenotipo en pacientes con la enfermedad COVID-19 y la diabetes, donde el fenotipo son rasgos físicos y conductas de un individuo. Contando con un conjunto de datos recopilado en los días 10 – 30 de marzo de 2020, llegando a 1317 registros. Para el análisis de datos los campos de edad y sexo fueron ajustados para el uso de regresión logística multi-variable. Obteniendo los resultados del análisis presentan que dentro del conjunto de datos un 10.6 % de personas fallecieron y un 18.0 % fue dado de alta a los 7 días, dejando al resto de datos con las características de una edad promedio de 69.8 $\pm$ 13.0 años, un IMC promedio de 28.4 y con una predominancia del 88.5 % de registros con diabetes tipo 2 donde complicaciones micro-vasculares y macro-vasculares fueron encontradas en 46.8 % y 40.8 % respectivamente. Llegando a la conclusión que las personas hospitalizadas por COVID-19, IMC, y sin control de glucosa a largo plazo, presentan positivo y son independientemente asociados con intubación o el fallecimiento dentro de 7 días.

Durante el año 2020 M. Nedyalkova *et al.* [19] reportaron una forma de clasificar un conjunto de datos de pacientes con diferentes enfermedades (DMT2, hipertonía arterial, enfermedad isquémica del corazón, entre otras) de manera diferente en comparación con lo visto anteriormente. Los autores hacen uso de la agrupación en clústeres conocido como *K-means* el cuál es usado para clasificar objetos en k grupos con características similares. El trabajo está enfocado en usar un método no enfocado en la identificación de los mejores descriptores para realizar la clasificación. También presentan el diseño de una metodología original la cual no intenta atacar el problema de los datos que faltan, si no que normaliza el conjunto de datos. La función de su metodología consiste en realizar una normalización de datos para posteriormente realizar una clasificación en tres grupos con al menos tres elementos dentro de cada grupo. El único contra de este trabajo es la cantidad de combinaciones posibles para generar el grupo perfecto llegando a un total de 24,360 combinaciones posibles y tener definido la cantidad de grupos que remplazara a la k dentro del algoritmo.

Saliendo de la rama de uso clínico de algoritmos clasificadores, en el 2017 Blanca Cuji *et al.* [20] presentaron un modelo predictivo de deserción estudiantil utilizando árboles de decisión utilizando datos recopilados a partir del año 2006 de la Universidad Técnica de Ambato correspondientes a los estudiantes de la carrera en informática. En donde utilizaron cinco variables socioeconómicas y cinco académicas, utilizando la herramienta *Rattle de R*, para realizar una construcción "top-down" del árbol de

decisión y una sección de datos 90/10 para su evaluación utilizando una curva Receiver Operating Characteristic (ROC) que alcanzaba un 94 % de efectividad en la predicción de los datos, llegando a la conclusión que existe el 29 % de probabilidad que personas casadas de etnia mestiza, casados, mayores a 35 años, con un promedio de notas de siete y en grados de inicio, deserten.

En el 2017 Yamilé *et al.* [2] realizó un estudio transversal con diseño muestral aleatorio, para detectar la preva-lencia de enfermedades crónicas no trasmisibles y sus factores de riesgo. El trabajo utilizó un total de 2085 registros de personas entre 14 municipios, de diferentes eda-des (32-56 años) utilizando variables como: sexo, edad, perímetro abdominal, gluco-sa, insulina, triglicéridos, colesterol, entre otros. Con el uso de medias y desviación estándar para generalizar los atributos presentaron las tablas a varios expertos del campo para diagnosticar cada registro. Llegando a la conclusión que a mayor edad (mayor de 50 años aproximadamente) se producen cambios hormonales y metabólicos que afec-tan a varios sistemas. Consecuentemente, desarrollando intolerancia a la glucosa, DMT2 y obesidad abdominal.

En este trabajo realizaremos un tratamiento a los datos que faltan a varios conjuntos de datos con el fin de utilizarlos en mejorar la clasificación diferentes algoritmos de clasificación (*TreeJ48*, *Naïve Bayes*, SMO y *Random Forest*). Posteriormente eliminar atributos no relevantes del conjunto utilizando diferentes métodos de compresión de datos como: PCA y NMF para realizar una comparativa de las clasificaciones resultantes, para demostrar que es factible el uso de tratamiento de datos y la reducción de atributos con algoritmos de compresión de datos sin perder una exactitud considerable en la clasificación.

Capítulo 3

Marco teórico

En este apartado abarcaremos ciertos conceptos que son importantes para la realización del trabajo, ya que cada uno de los siguientes apartados se complementan unos con otros.

3.1. Análisis de datos

El análisis de datos consiste en realizar operaciones en las cuales someteremos a los datos con la finalidad de alcanzar nuestros objetivos como, por ejemplo, encontrar patrones para ser interpretados como comportamiento de una persona. En general existen dos familias en el análisis de datos, las cualitativas y cuantitativas, estas modalidades utilizan conocimientos y técnicas completamente diferenciadas. La recolección de los datos y ciertos análisis preliminares pueden presentar problemas y dificultades que des-actualizarán la planificación establecida de cualquier proyecto.

En el 2018 Jesús García [21] menciona que, a finales del siglo XX la cantidad de datos que esta almacenada en bases de datos excedía la habilidad de los operadores para reducir y analizar los datos sin técnicas de análisis de datos automatizada. Afirmando que según informes de IBM el 90 % de los datos disponibles a la fecha del día de hoy han sido creadas en los últimos años. ¿A qué hacemos referencia cuando hablamos de datos?, ¿qué nos dice estos datos?, son las primeras preguntas que debemos de profundizar antes de realizar el análisis de estos, generalmente este enfoque se realiza principalmente a

los datos estructurados, que se encuentran en una gran variedad de formas como:

- Información en un formato de hoja de cálculo, donde cada columna puede ser un tipo de dato diferente.
- Matrices Multidimensionales.
- Tablas de datos relacionadas por llaves en columnas (SQL).
- El resultado de los datos que se ingresan en un formulario de un sitio web.
- Fichas estandarizadas de clientes.

Esta lista no es definitiva ya que, aunque no parezca, una gran variedad de conjuntos de datos puede ser transformados o estructurados para hacer el análisis más apropiado. En el caso de no ser posible, se pude extraer ciertas características del conjunto de datos para pasarlos a una estructura. Dentro del análisis de datos existen métodos para simplificar o transformar el contenido de los datos, donde destaca la escala de características.

3.1.1. Escala de características

También conocida como normalización de datos (data normalization) es el método utilizado para estandarizar un rango de propiedades de los datos, dado a que estos pueden tener una amplia variación entre sus datos, es necesario tratar la información para el uso de algoritmos de aprendizaje automático avanzados. Transformar la información en base a una escala donde los datos entran en el rango de valores de cero a uno es conocido como *min-max scaling*, que es obtenida de la ecuación 3.1:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{3.1}$$

Realizar un escalado es un paso primordial a la hora de implementar algoritmos como máquinas de soporte vectorial, dado a que utilizan la distancia de vectores entre dos puntos [22] y es más fácil calcular distancias si los valores se encuentran en un rango más pequeño como se ve en la **Figura 3.1** donde los valores máximos del eje de las

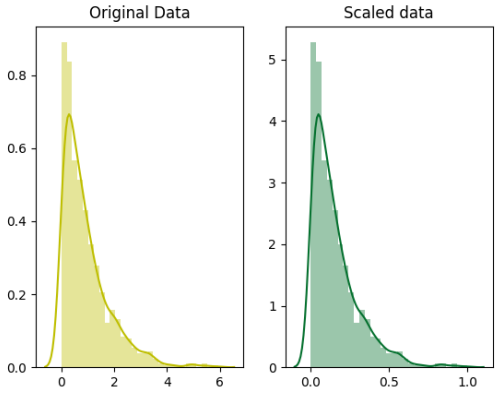


Figura 3.1: Min-Max scaling [22].

abscisas de los datos escalados (scaled data) no superan los valores de 1.0 manteniendo la misma forma que el gráfico amarillo a su izquierda (original data).

El propósito de la escala de características es cambiar la observación de los datos, uno de los métodos utilizados conocido como normalización describe a los datos con una distribución normal, también conocida como distribución Gaussiana, que es una curva en forma de campana. Teniendo una distribución estadística especifica donde las observaciones aproximadamente iguales caen por encima y debajo de la media (**Figura 3.2**).

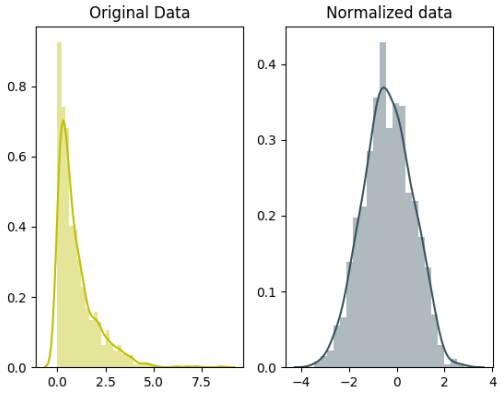


Figura 3.2: Normalización de datos [22].

Otro método utilizado es la estandarización o puntuación z. La cual transforma los datos de tal manera que la distribución tenga una media de cero y una desviación

estándar de uno (similar a la campana generada con la normalización), que es ampliamente usado en redes neuronales, regresión logística y máquinas de soporte vectorial. La Normalización es otra manera de tratar a los datos, ya que en este método se esta cambiando la forma en la que se distribuyen los datos. De la misma manera que la escala de características la normalización y estandarización de datos ayuda a realizar de manera más efectiva los cálculos matemáticos realizados por algoritmos avanzados como máquinas de soporte vectorial, redes neuronales y regresión logística.

3.2. Inteligencia artificial

La inteligencia artificial o IA, es la inteligencia llevada a cabo por máquinas. En ciencias de la computación, una máquina inteligente ideal es un agente flexible que percibe su entorno y lleva a cabo acciones que maximicen sus posibilidades de éxito en algún objetivo o tarea [23]. Coloquialmente, el término inteligencia artificial se aplica cuando una máquina imita las funciones cognitivas que los humanos asocian con otras mentes humanas, como, por ejemplo: percibir, razonar, aprender y resolver problemas.

La definición de IA es la capacidad de un sistema para interpretar correctamente datos externos, para aprender de dichos datos y emplear esos conocimientos para lograr tareas y metas concretas a través de la adaptación flexible. En el 2007 según Takeyas, la IA componía una de las ramas encargadas de estudiar los modelos de cómputo capaces para ser capaces de realizar actividades propias como seres humanos basándose en dos características primordiales: el razonamiento y la conducta, creando varios tipos de sistemas que fueron categorizados por Stuart Russell y Peter Norvig de la siguiente manera:

- **Sistemas que piensan como humanos:** Los cuales tratan de imitar el pensamiento humano, como por ejemplo las redes neuronales.
- **Sistemas que actúan como humanos:** Estos sistemas tratan de actuar como humanos o el comportamiento de los humanos, el mejor ejemplo de este caso es la robótica.
- **Sistemas que piensan racionalmente:** Como su nombre dice utilizan la lógica para tratar de imitar el pensamiento racional de un ser humano, un sistema experto que puede darte motivos de su elección es un gran ejemplo de este caso.
- **Sistemas que actúan racionalmente:** Donde los sistemas intentan emular de forma racional el comportamiento humano.

En IA existe un acuerdo sobre cuáles son los resultados atribuibles a cierta rama de la informática, así como la clasificación de sus métodos y técnicas desarrolladas, logrando abarcar cuatro temas en general [24].

- Resolución de problemas de búsqueda.
- Representación de conocimiento y sistemas basados en conocimiento.
- Aprendizaje automático.
- Inteligencia artificial distribuida.

¿Y esto en qué nos ayuda?, bueno es una pregunta difícil de contestar, ya que la IA puede llegar a tener muchas aplicaciones diferentes, entre ellas ha destacado en aplicaciones en los juegos de mesa o virtuales como: Damas, Ajedrez o Go, donde la cantidad de combinaciones o variaciones de estados de juego son inmensas, haciendo que encontrar la mejor ruta (o mejor jugada) a tomar fuera la mejor selección que un humano podría hacer.

También en el ámbito de robótica se han encontrado avances como con robots de exploración y reconocimiento que han sido enviados a Marte para recolectar información de este planeta y no depender de acciones humanas para su funcionamiento. Incluso pueden ser elementos de la vida cotidiana como robots bípedos, los cuales son capaces de caminar, correr y reconocer objetos móviles, posturas y gestos a partir de información provista por la amplia gama de cámaras y sensores. Esto también incluye a los coches autónomos por la empresa Tesla o el coche *Stanley* que fue el ganador de la carrera *DARPA Grand Challenge* en el 2005 recorriendo 212.4 Km sin un conductor. La informática no ha dejado de avanzar desde sus inicios hace casi 80 años.

3.3. Aprendizaje automático

El aprendizaje automático es la rama de la ciencia tiene como objetivo el desarrollar técnicas que permitan a las computadoras aprender. Para ser más precisos, tratar de crear algoritmos capaces de generalizar el comportamiento y reconocer patrones a partir de información suministrada en forma de ejemplos. En muchas ocasiones el proceso de aprendizaje automático se solapa con el de minería de datos (*Data Mining*) ya que ambas disciplinas están enfocadas en el análisis de datos. Ambos sistemas buscan entre

los datos para encontrar algún patrón, sin embargo, en lugar de extraer datos para la compresión humana, el aprendizaje automático utiliza esos datos para ajustar las acciones del programa que está ejecutando.

A un nivel básico podríamos decir que una de las tareas del aprendizaje automático es extraer el conocimiento de propiedades no observadas de un objeto basándose en ciertas propiedades que sí se han observado en el objeto, o en predecir un comportamiento de acuerdo a lo que ha ocurrido en el pasado. El aprendizaje dentro del contexto humano es el proceso a través del cual se adquieren habilidades, destrezas, conocimientos, conductas o valores como resultado del estudio, experiencia, intuición, razonamiento y observación. En el aprendizaje automático se considera aprendizaje a aquello que la máquina pueda aprender a partir de la experiencia, no a partir del reconocimiento de patrones programados a priori [25].

Normalmente, cuando se aborda un nuevo problema relacionado con aprendizaje automático es necesario saber qué tipo de objetos va a predecir, entrando en una de las clases habituales como:

- **Regresión**: En donde se intenta predecir un valor real. Como predecir la nota de un estudiante en el examen final, basándose en las calificaciones de diversas tareas realizadas en el curso.
- **Clasificación** **(binaria o multi-clase)**: Las cuales intentan predecir la clasificación de objetos sobre un conjunto de clases prefijadas. Como determinar la clasificación de una noticia (deporte, ciencia, política, entretenimiento, etc).
- **Ranking**: Aquí se intenta predecir el orden óptimo de un conjunto de objetos según el orden de relevancia predefinido. Como un buscador que devuelve recursos de internet como respuesta de una búsqueda.

En la **Figura 3.3** podemos observar un ejemplo a grandes rasgos de la clasificación (binaria) donde una recta (hiper-plano) separa a un conjunto de puntos en base a diferentes valores. Los puntos que se encuentran a la izquierda del hiper-plano son objetos con resultado de clase *verde* y los restantes entran en la clase *azul*

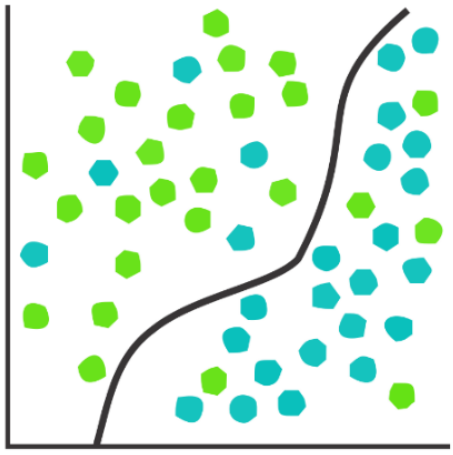


Figura 3.3: Ejemplo básico de clasificación binaria [25].

Dependiendo del tipo de salida que se produzca, los algoritmos de aprendizaje automático se pueden agrupar en:

- **Aprendizaje supervisado:** Establece una correspondencia con una función entre las entradas y salidas deseadas del sistema donde la base de conocimientos del sistema está formada por ejemplos etiquetados (donde se conoce la clasificación correcta).
- **Aprendizaje no supervisado:** Aquí se lleva a cabo un modelado sobre un conjunto de datos formado únicamente por entradas del sistema, en otras palabras, no conocemos la clasificación de estas entradas. Por lo que se busca que el sistema sea capaz de reconocer patrones para etiquetar cada una de las entradas.
- **Aprendizaje semi-supervisado:** Es una combinación de los algoritmos anteriores, teniendo en cuenta ejemplos clasificados y no clasificados.
- **Aprendizaje por refuerzo:** En este tipo de algoritmos se aprende con base a la observación del mundo y con un flujo de información en dos direcciones, entre mundo-máquina y viceversa, realizando un proceso de *prueba y error* reforzando a las acciones que reciben una respuesta positiva en el mundo.

- **Transducción:** Algo similar al aprendizaje supervisado, pero el objetivo es únicamente tratar de predecir las categorías en las que caen los ejemplos basándose en los ejemplos de entrada.
- **Aprendizaje multi-tarea:** Este tipo de algoritmo engloba a todos aquellos que usan conocimiento previamente aprendido por el sistema de cara a enfrentarse a problemas parecidos a los ya vistos.

También hay que delimitar el tema separando al aprendizaje automático del aprendizaje profundo o conocido como *Deep Learning*, ya que el aprendizaje profundo está enfocado en la combinación de avances en cómputo con redes neuronales especiales para aprender patrones complicados en grandes cantidades de datos. Los métodos utilizados han sido utilizados para identificar objetos en imágenes y palabras en sonidos.

3.4. Clasificadores

El termino clasificador es una referencia al algoritmo utilizado asignar un elemento no etiquetado en una categoría concreta. Este tipo de algoritmos nos permite ordenar o disponer por clases a los datos, a partir de información que los caracteriza. Para la implementación de cualquier clasificador es necesario tener una serie de características concretas, pero la adición de parámetros irrelevantes hace más difícil la tarea de clasificar los datos. Esta sección tiene como objetivo presentar los conceptos básicos junto con una breve explicación de cada clasificador utilizado dentro del trabajo.

3.4.1. Árboles de decisión

El algoritmo de árboles de decisión tiene como objetivo crear un modelo que prediga el valor de una variable utilizando aprendizaje inductivo a partir de observaciones y construcciones lógicas aprendiendo reglas de decisión (*si-entonces-si no*) inferidas de las características de los datos, generalmente es compuesto por dos etapas.

En la primera etapa se construye el árbol a partir de un conjunto de datos de entrenamiento en donde cada nodo interno se compone de un atributo de prueba y porción del conjunto de entrenamiento presente en el nodo es dividida de acuerdo con los valores

del atributo. Cuando en un nodo se tienen objetos de más de una clase se genera un nodo interno; cuando contiene objetos de una clase, se genera una hoja con la asignación de la clase. Cuanto más profundo es un árbol, más complejas son las reglas de decisión y el modelo se vuelve más ajustado.

En la segunda etapa cada objeto nuevo es clasificado recorriendo desde el nodo raíz hasta una hoja generando un camino basado en las decisiones de los nodos internos en base a los datos utilizados en el entrenamiento del árbol [26].

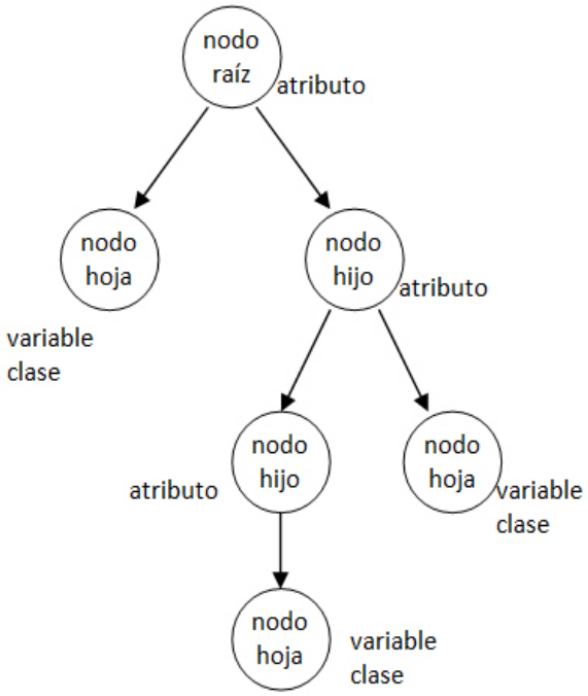


Figura 3.4: Estructura de un árbol de decisión [26].

Las ventajas del árbol de decisión son:

- Los árboles se pueden visualizar, por lo que los hace sencillos de entender e interpretar.
- Requiere poca preparación de datos. Para casos específicos a menudo requieren de normalización, hay que tener en cuenta que no acepta valores que faltan.
- El costo de usar un árbol para predecir es logarítmico.
- Capaz de manejar datos numéricos como categóricos.

- Utiliza un modelo de caja blanca, donde puede ser explicado con lógica booleana.
- Es posible validar un modelo con pruebas estadísticas.
- Funciona incluso si el modelo real a partir del cual se generaron los datos viola sus suposiciones.

Las desventajas de los árboles de decisión incluyen:

- Pueden crear árboles demasiado complejos que no generalizan bien los datos, conocido como *overfitting*, el cual es tratado con un factor de poda o un número mínimo de muestras necesarias para la creación de un nodo o hoja
- Las predicciones de los árboles de decisión no son uniformes ni continuos, sino aproximaciones constantes por partes como en la **figura 3.5**.
- Los algoritmos prácticos de aprendizaje del árbol de decisiones se basan en algoritmos heurísticos, dado a que el problema de aprender un árbol de decisiones es un problema *NP*-completo.
- Hay conceptos que son difíciles de aprender porque los árboles de decisión no los expresan fácilmente, como la compuerta *XOR*.
- Se puede crear un árbol sesgado si alguna clase es dominante.

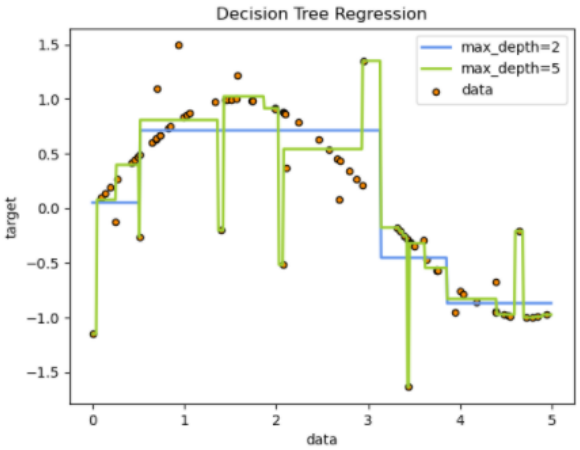


Figura 3.5: Ejecución de un árbol de decisión [26].

3.4.2. Naïve Bayes

El clasificador *Naïve Bayes* (CNB), es un algoritmo de aprendizaje simple pero potente, ya que este se base en la probabilidad condicional y el teorema de Bayes. Pero antes de hablar del algoritmo hay que entender que es la probabilidad condicional para utilizar el teorema de Bayes junto con la regla de la cadena y calcular dicha probabilidad.

Suponemos que tenemos un dado con seis caras, ¿Cuál es la probabilidad de obtener un seis al lanzar el dado?, bueno la respuesta es fácil, $\frac{1}{6}$. Existen seis resultados posibles e igualmente probables, pero solo uno de ellos es de interés. Pero, ¿Qué pasaría si ya se lanzó el dado y el resultado es un número par? ¿Qué probabilidad existe ahora?

Ahora los resultados son reducidos a tres, por lo que la probabilidad es mayor $\frac{1}{3}$. En el primer caso, no se tenía la información previa del resultado, por lo que se necesitaba considerar más resultados posibles. Calcular la probabilidad de un evento A, dada la ocurrencia de un evento B es denominado probabilidad condicional de A dado B, o denotado $P(A|B)$, en este caso $P(Seis|Par) = \frac{1}{3}$. Para calcular la probabilidad de un evento A dada la ocurrencia de B se utiliza la ecuación 3.2.

$$P(A|B) = \frac{P(A,B)}{P(B)} \tag{3.2}$$

Dónde $P(A,B)$ denota la probabilidad de que A y B sucedan al mismo tiempo y $P(B)$ denota la probabilidad de que suceda B. Por la formula anterior, la probabilidad de B no puede ser menor a cero, porque no tiene sentido hablar de la probabilidad de A dado B si la ocurrencia con B no es posible.

Por lógica se puede calcular la probabilidad de A, dada la ocurrencia de múltiples eventos (B_1, B_2, ..., B_N) de la siguiente forma:

$$P(A|B_1, B_2, ..., B_n) = \frac{P(A, B_1, B_2, ..., B_n)}{P(B_1, B_2, ..., B_n)} \tag{3.3}$$

Y Aquí es donde entra el denominado teorema de Bayes, invirtiendo el orden de las ocurrencias de los eventos (suponiendo que ha ocurrido el evento A y se tiene que calcular el problema del evento B y enfocarse solamente en el numerador):

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{3.4}$$

$$P(A|B_1, B_2, ..., B_n) = \frac{P(B_1, B_2, ..., B_n)P(A)}{P(B_1, B_2, ..., B_n)} \tag{3.5}$$

Por último, la independencia, donde decimos que los eventos A y B son independientes si $P(A|B) = P(A)$. Lo que significa que el evento A no se ve afectado por la ocurrencia del evento B, por lo que se puede decir que la ocurrencia de A y B al mismo tiempo es igual al producto de los problemas de los eventos A y B que ocurren por separado (Ecuación 3.6).

$$P(A, B|C) = P(A|C)P(B|C) \tag{3.6}$$

Ahora, el clasificador es un vector de n características donde se intenta determinar la clase del vector a partir de un conjunto de k clases y_1,y_2,. . . ,y_k. Como un ejemplo, determinar si el día de hoy lloverá. Existen dos clases posibles ($k = 2$): llueve y no llueve, y la longitud del vector estará dada por la cantidad de características, en este caso tres ($n = 3$), donde la primera característica hace referencia a si está nublado o soleado, la segunda en base a la humedad (alta o baja) y por último a la temperatura (alta, media, baja). Obteniendo vectores de la siguiente manera:

$$[Nublado, H_{alta}, T_{baja}], [Soleado, H_{baja}, T_{media}], [Nublado, H_{baja}, T_{alta}] \tag{3.7}$$

Aplicando un poco de razonamiento y lógica obtenemos las ecuaciones 3.8 y 3.9.

$$L = P(Llueve|Nublado, H_{alta}, T_{baja}) \tag{3.8}$$

$$NL = P(NoLlueve|Soleado, H_{baja}, T_{alta}) \tag{3.9}$$

Donde si el resultado de L es mayor a NL se asume que el vector entra en la clase llueve, pero para calcular las probabilidades es necesario tener un conjunto de ejemplos previamente etiquetados de la siguiente manera:

...

$$[Nublado, H_{alta}, T_{baja}] \rightarrow Llueve$$

$$[Nublado, H_{alta}, T_{baja}] \rightarrow Llueve$$

...

Suponiendo que se tiene que calcular el vector $[Nublado, H_{baja}, T_{baja}]$ obtendríamos una expresión parecida a la siguiente:

$$P(Llueve|Nublado, H_{baja}, T_{baja}) = \frac{P(Nublado|H_{baja}, T_{baja}, Llueve)P(H_{baja}|T_{baja}, Lueve)P(T_{baja}|Lueve)P(Llueve)}{P(Nublado, H_{baja}, T_{baja})} \tag{3.10}$$

Pero hacer cálculos para todas las clases posibles es muy costoso y lento, por lo que se necesita hacer suposiciones sobre el problema para que los datos se simplifiquen. Los clasificadores ingenuos de Bayes asumen que todas las características son independientes entre sí, permitiéndonos reescribir la expresión aplicando el teorema de Bayes y asumiendo la independencia entre cada par de características de la siguiente manera:

$$P(Llueve|Nublado, H_{baja}, T_{baja}) = \frac{P(Nublado|Llueve)P(H_{baja}|Lueve)P(T_{baja}|Lueve)P(Llueve)}{P(Nublado, H_{baja}, T_{baja})} \tag{3.11}$$

Donde el conteo se puede calcular de forma más rápida las probabilidades como $P(Nublado|Llueve)$ contando el número de entradas que se clasifican como *Llueve* y presentaron un atributo *Nublado*. El algoritmo es llamado *ingenuo* dado a la suposición de independencia que se crea entre las funciones la mayor parte del tiempo.

3.4.3. Random forest

La implementación del clasificador *Random forest* solventa una de las problemáticas causadas por los arboles de clasificación conocido como *overfitting*, que ocurre cuando un modelo flexible comienza a memorizar los datos de entrenamiento, perdiendo la varianza del conjunto de datos. Además, el *overfitting* causa problema en el caso contrario (un modelo inflexible) donde se hacen asunciones del conjunto de datos, lo cual provoca que el modelo no sea eficiente al generalizar nuevos datos. El algoritmo *Random forest* es compuesto por dos partes: La primera parte es un conjunto de árboles de decisión, el cual es llamado *bosque* (forest); La segunda parte, recibe el nombre de *aleatorio* (*random*) por la creación de muestreos y sub-conjuntos aleatorios del conjunto de datos basándose en el conjunto de datos original.

Random forest combina cientos o miles de árboles de decisión, entrenando a cada uno de ellos con un conjunto de datos ligeramente diferente dividiendo los nodos de cada árbol, limitando el número de características disponibles, la cantidad de registros disponibles. Logrando que las predicciones finales del bosque aleatorio se realicen promediando las predicciones de cada árbol individual.

Para comprender mejor el funcionamiento del algoritmo es mejor ver un escenario: "Se debe decidir si las acciones de la automovilística Tesla subirán, se tiene acceso a una docena de analistas que no tienen conocimiento previo de esta compañía. Cada analista tiene un sesgo (*bias*) bajo, por lo que no entra con ninguna suposición y se le permite aprender de un conjunto de datos de informes y noticias."

Esto puede parecer una situación ideal para el algoritmo *Random forest*, pero el problema radica en la posibilidad que los informes contengan ruido o información no requerida. Dado a que los analistas basan sus predicciones completamente de los datos (altamente flexibles), pueden ser influenciados por información irrelevante. La solución radica en no depender de ningún individuo, si no poner en común los votos de cada analista. Además, cada analista accede a una sección de la información, esperando que este muestreo anule la información irrelevante. En la vida real confiamos en múltiples fuentes y, por lo tanto, *Random forest* es una manera intuitiva de predecir y una mejora a los clasificadores de árboles.

3.4.4. Máquinas de soporte vectorial

Las máquinas de soporte vectorial (por sus siglas en inglés SVM) son un algoritmo de clasificación y regresión desarrollado en la década de los 90, dentro del campo de la ciencia computacional. Originalmente era utilizado para la clasificación binaria, pero su aplicación se ha extendido a la clasificación múltiple y regresión. Este algoritmo se fundamenta en el máximo margen de clasificación que se basa en el concepto de hiper-plano al mismo tiempo.

El hiper-plano se define como un sub-espacio plano y afín de dimensiones $p-1$. El término afín significa que el sub-espacio no tiene que pasar por el origen. En un espacio de dos dimensiones, el sub-espacio seria de una dimensión (una recta). La definición matemática de un hiper-plano de dos dimensiones se describe como la ecuación de la recta (**Ecuación 3.12**):

$$B_0 + B_1 x_1 + B_2 x_2 = 0 \tag{3.12}$$

La **figura 3.6** presenta un hiper-plano de un espacio bidimensional donde la ecuación que describe al hiperplano es: $1+2x_1+3x_2 = 0$. Donde la región de color azul representa el espacio que se encuentran los puntos para los que $1 + 2x_1 + 3x_2 > 0$ es verdadera y la zona roja el caso contrario.

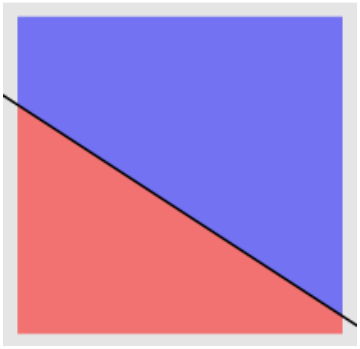


Figura 3.6: Hiper-plano de separación en dos dimensiones [27].

La clasificación binaria sucede cuando se puede separar de manera perfecta y linealmente a dos clases (+1 y -1) de un conjunto de datos, donde el hiper-plano de separación cumple las ecuaciones 3.13 y 3.14:

$$B_0 + B_1 x_1 + ... + B_p x_p > 0, si, y_i = 1 \tag{3.13}$$

$$B_0 + B_1 x_1 + ... + B_p x_p < 0, si, y_i = -1 \tag{3.14}$$

La definición para casos perfectamente separables linealmente nos da un número infinito de posibles hiper-planos (**figura 3.7**), lo que hace necesario un método para seleccionar uno de estos hiper-planos como el clasificador óptimo.

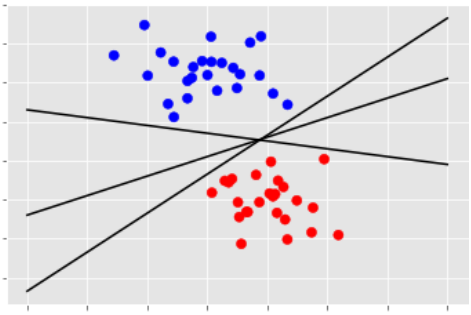


Figura 3.7: Tres posibles hiper-planos de un caso perfectamente separable [27].

Para la solución de este problema se selecciona el hiper-plano que se encuentra más alejado de todas las observaciones del entrenamiento. Esto es conocido como "hiper plano óptimo de separación". Para identificar el hiper-plano, se tiene que calcular la distancia perpendicular de cada observación a un determinado hiper plano. La menor de estas distancias determina cuan alejado está de las observaciones del entrenamiento. La **figura 3.8** presenta el hiper-plano óptimo de separación con una línea negra continua y su margen con líneas punteadas del mismo color. Estas líneas punteadas se les conoce como vectores de soporte y cualquier modificación conlleva a cambios en el hiper-plano de separación, lo que conlleva a nuevos vectores de soporte.

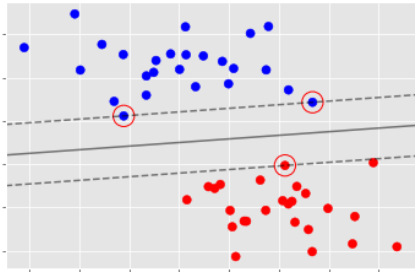


Figura 3.8: Hiper-plano óptimo de separación [27].

Las máquinas de soporte vectorial consiguen buenos resultados cuando el límite de la separación entre clases es aproximadamente lineal. En dado caso de no ser lineal, decae drásticamente. Una estrategia para enfrentarse a escenarios en donde la separación de los grupos es de tipo no lineal consiste en expandir las dimensiones del espacio original (**Figura 3.9**).

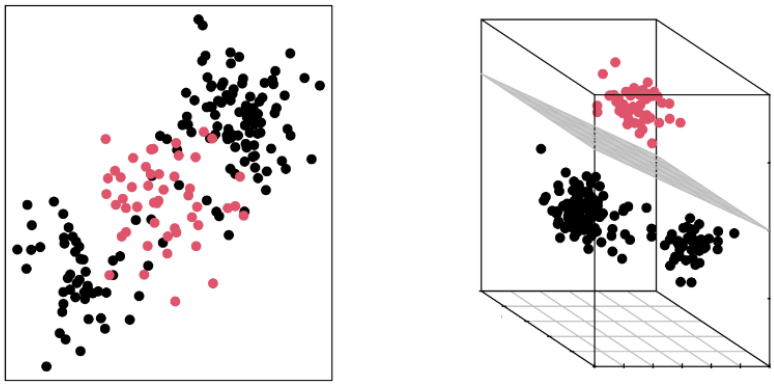


Figura 3.9: Escenarios de tipo no lineal [27].

Dado al incremento de dimensiones, la máquina de soporte vectorial participa en una matriz n x n, donde n es el número de observaciones de entrenamiento. Por esta razón, lo que más influye en el tiempo de computo necesario para entrenar depende del número de observaciones y no en el de predictores.

3.5. Big data

Gracias a la gran cantidad de datos disponibles, generados de manera continua e incesante por diferentes entidades, dispositivos, usuarios e incluso servicios ha implicado en el desarrollo de nuevos métodos científicos para tener sistemas y procedimientos capaces de almacenar, procesar y analizar dichos datos.*Big Data* es un término cuya traducción equivale a "datos masivos", dado a que el tamaño de estos datos supera considerablemente la capacidad de captura, almacenamiento, gestión y análisis del *software* utilizado en bases de datos. Por ejemplo, en Facebook se suben 10 millones de fotos por hora y esos datos son almacenados frecuentemente en bases de datos sin ser alterados o tocados.

La ciencia de datos nos revela tendencias y genera información que las empresas pueden utilizar para tomar mejores decisiones a la hora de crear productos o servicios innovadores. La ciencia de datos combina múltiples campos que incluyen estadística, métodos científicos y análisis de datos para extraer el valor de los datos. Y la pregunta *¿por qué es tan importante?* destaca frecuentemente, con la tecnología moderna la creación y almacenamiento de datos son mayores con el paso del tiempo.

Pero el concepto de *Big Data* no solamente hace referencia al tamaño de la información, sino también a la variedad del contenido y a la velocidad que se generan, conocidas también como las *3V* (volumen, velocidad y variedad de datos). Algunos autores incluyen una cuarta y quinta *V* (*Value* y *Veracity*) [28], para hacer referencia a la veracidad de los datos, la cual es vital para un negocio dado al valor de la información que presentan los análisis. Una empresa define a las $3V$ de la siguiente manera:

- **Volumen:** Corresponde al gran volumen de datos que se generan diariamente en las empresas y organizaciones de todo el mundo.
- **Velocidad:** Se trata de los flujos de datos, la creación de registros estructurados y disponibilidad para el acceso y la entrega.
- **Variedad:** Capacidad de combinar una gran variedad de información digital en los diferentes formatos en los que se puedan presentar.

Para comprender de una mejor manera a la ciencia de datos tenemos que tener la compresión de otros términos relacionados con el campo, como el caso de la inteligencia artificial (IA) y aprendizaje automático. Usualmente estos términos son usados indistintamente, pero la ciencia de datos es un subconjunto de la IA que se refiere a las áreas de las estadísticas, los métodos científicos y análisis de datos. Mientras que el aprendizaje automático consiste en las técnicas que permiten a las computadoras el descubrimiento de cosas o conocimiento a partir de los datos. La ciencia de datos es un proceso iterativo donde su ciclo de vida está compuesto por seis etapas principales como la planificación, construcción de un modelo de datos, evaluación del modelo, explicación del modelo, implementación y monitorización del modelo.

3.5.1. Análisis de componentes principales

El análisis de componentes principales conocida como ACP (o por sus siglas en inglés PCA) trata de explicar la estructura de las varianzas y covarianzas de un conjunto de

variables, mediante combinaciones lineales de ellas llamadas componentes principales. PCA aspira a reducir o simplificar los datos para tener un análisis e interpretación de una forma sencilla [29], que es de gran ayuda en los campos como minería de datos, análisis numérico y aprendizaje automático. También es utilizada en otros campos como seguridad informática, análisis de imágenes e investigación genética.

El ejemplo más conocido dentro del ambiente de PCA, es en el ámbito de reconocimiento de patrones utilizando la base de datos nombrada *Iris* por *R.A Fisher* en 1936 [30], la cual fue recolectada por Edgar Anderson con el fin de demostrar que las medidas presentadas pueden utilizarse para diferenciar a las diferentes especies de plantas iris. El conjunto de datos está compuesto de 3 clases de 50 casos cada una donde cada clase se refiere a un tipo de planta iris, donde contiene los siguientes atributos:

- Longitud del sépalo (sepal) en cm.
- Ancho del sépalo en cm.
- Longitud del pétalo (petal) en cm.
- Ancho del pétalo en cm.
- Clase (Species):
  - Setosa.
  - Versicolour.
  - Virginica.

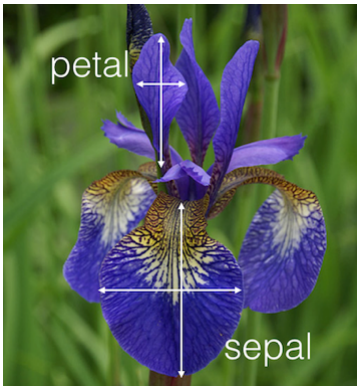


Figura 3.10: Pétalo (petal) y sépalo (sepal) de una flor [31].

La forma generalizada para aplicar algoritmos PCA, pueden ser divididos en cinco pasos que se enlistan a continuación:

1. Cargar datos.
2. Normalizar datos.
3. Obtener auto-vectores y auto-valores a partir de la matriz de covarianza.
4. Seleccionar auto-vectores correspondientes por cada componente principal.
5. Proyectar el conjunto de datos original sobre el nuevo espacio de dimensión

En la parte de carga de datos es necesario leer de manera eficiente para que no causen una carga al dispositivo que se esté utilizando, los datos generalmente están representados por una tabla donde cada columna representa uno de los componentes principales obteniendo una representación como se muestra en la **Tabla 3.1**.

Tabla 3.1: Representación del conjunto de datos [31].

	Lng. sépalo	**Anch. sépalo**	**Lng. pétalo**	**Anch. pétalo**	Especie
145	6.7	3.0	5.2	2.3	Iris-Virginica
146	6.3	2.5	5.0	1.9	Iris-Virginica
147	6.5	3.0	5.2	2.0	Iris-Virginica
148	6.2	3.4	5.4	2.3	Iris-Virginica
149	5.9	3.0	5.1	1.9	Iris-Virginica

Cuando las distintas características de un conjunto de datos están expresadas en distintas escalas lo cual pone una limitante a los algoritmos de PCA, por ello siempre se suele hacer la normalización de sus valores, en estos casos se utiliza una distribución gaussiana o normal, las cuales se encuentran implementadas en diferentes librerías de varios lenguajes de programación. También se suelen utilizar transformaciones ortogonales (rotaciones) de las variables originales.

Para la generación de auto-vectores (**Figura 3.11**), que son las direcciones que la varianza de datos es mayor, respetan la esencia principal de la información contenida en un conjunto de datos, mientras que el auto-valor es un número que representa el valor de la varianza sobre el auto-vector. Para encontrar los componentes principales que condensen la esencia de información del conjunto de datos en necesario calcular la matriz de covarianza, que nos da la medida de dispersión conjunta entre variables y en base a la matriz sacamos los auto-vectores y auto-valores.

```
Eigenvectors
[[ 0.52237162 -0.37231836 -0.72101681  0.26199559]
 [-0.26335492 -0.92555649  0.24203288 -0.12413481]
 [ 0.58125401 -0.02109478  0.14089226 -0.80115427]
 [ 0.56561105 -0.06541577  0.6338014   0.52354627]]

Eigenvalues
[ 2.93035378  0.92740362  0.14834223  0.02074601]
```

Figura 3.11: Auto-vectores y auto-valores.

Para seleccionar los auto-vectores correspondientes, hay que recordar que los auto-valores son una medida de la varianza de los datos. Para ello, se usa una métrica conocida como varianza explicada, la cual muestra cuánta varianza se puede atribuir a cada uno de los auto-vectores.

En la **Figura 3.12** se puede apreciar que la mayor parte de la varianza corresponde al primer componente por un 70 % y el segundo componente por un 20 % aproximadamente, mientras que la tercera y cuarta parte pueden ser descartadas ya que estos dos componentes solo explican el aproximadamente el 10 % de la varianza.

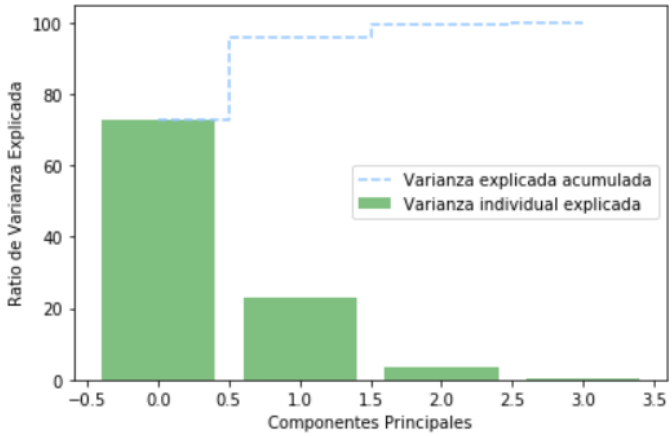


Figura 3.12: Varianza explicada.

Una vez seleccionados los auto-vectores correspondientes simplemente hay que realizar la conversión de dimensiones y representar la información en un nuevo espacio, en este caso se empieza con un conjunto de datos de cuatro dimensiones y se termina con un conjunto de datos de dos (**Figura 3.13**).

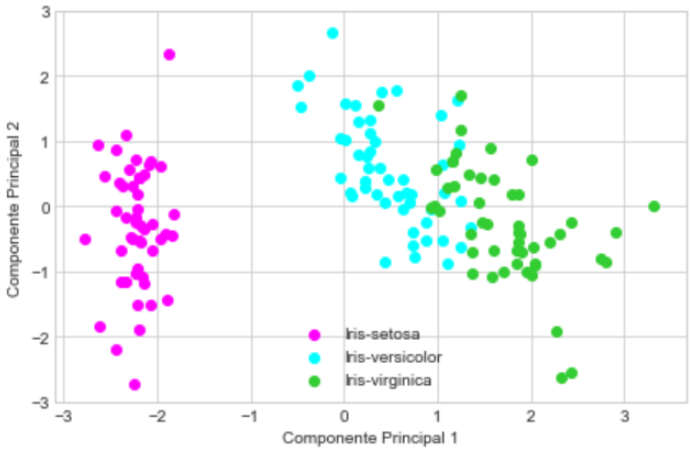


Figura 3.13: Representación gráfica del conjunto de datos en dos dimensiones.

3.6. Procesamiento de lenguaje natural

Es un campo de las ciencias de la computación, inteligencia artificial y lingüística que estudia las interacciones entre computadoras y el lenguaje humano. El procesamiento de lenguaje natural o PLN está enfocado en la formulación e investigación de mecanismos eficaces computacional-mente para la comunicación entre máquinas y personas mediante lenguaje natural. Hasta la década de 1980 la mayoría de los sistemas PLN se basaban en un conjunto de reglas para el análisis de un texto, pero obviamente existían ciertos problemas, *Edwar Sapir* autor de la frase *"All grammars leak"* dio a conocer que no era posible proveer una caracterización completa de frases bien formadas dado a que la gente siempre encontraba la forma de doblegar a las reglas para lograr su comunicación.

Esto fue generalizado en tres dificultadas para lograr el procesamiento del lenguaje natural, las cuales son:

- Ambigüedad
- Separación entre palabras.
- Recepción imperfecta de datos.

El objetivo de una ciencia lingüística es caracterizar y explicar la multitud de observaciones lingüísticas que nos rodean, en conversaciones, escritos y otros medios. Al tratar

de manera computacional una lengua implica usar un proceso de modelización matemática. Los lingüistas computacionales se encargan de preparar un modelo lingüístico, que generalmente se separan en dos aproximaciones:

- **Modelo Lógico** (gramáticas)**:** Donde se escriben reglas de reconocimiento de patrones estructurales, empleando formalismo gramatical concreto. Estas reglas definen los patrones que hay que reconocer para resolver la tarea (traducción de textos, búsqueda de información, etc.).
- **Modelo probabilístico del lenguaje natural** (basado en datos)**:** Se recopilan colecciones de ejemplos y datos llamados corpus y a partir de ellos se calculan frecuencias de diferentes unidades lingüísticas (letras, palabras, oraciones) y su probabilidad de aparecer en un contexto determinado.

Algunos de los componentes más relevantes del procesamiento de lenguaje natural son:

- **Análisis morfológico:** Consiste en el análisis interno de las palabras que forman oraciones para extraer lemas, rasgos flexivos, unidades léxicas compuestas.
- **Análisis sintáctico:** consiste en el análisis de la estructura de las oraciones de acuerdo al modelo gramatical empleado (lógico o probabilístico).
- **Análisis semántico:** Proporciona interpretación de oraciones. Una vez que las ambigüedades son eliminadas.
- **Análisis pragmático:** Incorpora el análisis del contexto de uso a la interpretación final.

3.6.1. Recuperación de información

Es la ciencia de la búsqueda de información en documentos electrónicos y cualquier tipo de colección documental digital, encargada de la búsqueda dentro de estos mismos, búsqueda de meta-datos que describan documentos. También es el conjunto de tareas mediante las cuales el usuario localiza y accede a los recursos de información que son pertinentes para la resolución del problema planteado. Algunos de los recursos utilizados en la recuperación de la información son:

- Bases de datos.
- Internet.
- Agentes inteligentes.
- Palabras calve.
- Tesauros.
- Ecuaciones de búsqueda.

También, cuenta con diferentes técnicas y herramientas, entre ellas se encuentran las más importantes a destacar que son:

- **Recuperación lógica difusa:** Nos permite formular consultas con frases normales y luego la maquina procesa las palabras que considera relevantes, basándose en proposiciones lógicas con valores de verdadero y falso, teniendo en cuenta la localización del documento.
- **Ponderación de términos:** Definimos un valor adecuado a los criterios de búsqueda, dependiendo de los intereses del usuario, por lo tanto, la recuperación depende.
- **Clustering:** Donde se crea un modelo probabilístico que utiliza la frecuencia de los términos de búsqueda en un documento recuperado.
- **Stemming:** En esta técnica se elimina las posibles confusiones semánticas que se puedan dar en la búsqueda de un concepto.
- **Lingüísticas:** Pretenden acortar de una manera eficaz los documentos relevantes.

3.7. Paquetes esenciales de Python

En esta sección se dará una breve explicación de los paquetes y herramientas utilizados en el trabajo realizado.

3.7.1. Pandas

Desde su desarrollo en el 2008 Pandas ha sido uno de los paquetes libres que ha sido utilizado para el manejo de datos dentro del entorno de la ciencia de datos (en el 2012 saco su edición especializada en análisis de datos), gracias a ciertas características especiales que maneja dentro del entorno de *Python* entre ellos una re-dimensión de conjuntos de datos, indexado y manejo de subconjuntos. También ofrece ayuda para exportar información estructurada de diferentes archivos entre ellos archivos de texto (*CSV's*), *Excel*, bases de datos *SQL* y el formato HDF5. Cierta parte de la paquetería esta optimizada en el rendimiento que está escrito en Cython [32].

3.7.2. PyDoxc

PyDocx es el paquete que nos permite acceder a archivos *MS Word* (Office Open *XML*) escritos a partir del 2006 en adelante utilizando la extensión *docx*, utilizada en los días actuales por la paquetería de *Microsoft Word*, ya que este tipo de ficheros utilizan un lenguaje similar al hipermercado conocido como *HTML* (etiquetas, referencias y recursos) [33].

3.7.3. BeautifulSoup

BeautifulSoup es la librería que fue nombrada a partir del poema de *Lewis Carroll* utilizado en Alicia en el país de las maravillas cantada por el personaje conocido como *tortuga falsa* (*Mock Turtle*). Tal como su homónimo en el país de las maravillas, BeautifulSoup intenta darle sentido a lo que no tiene sentido, ya que nos ayuda a formatear y organizar contenido web desordenado arreglando el formato *HTML* y presentarnos dicha información en objetos de Python que pueden ser transitables a representarse en diferentes estructuras, generalmente estas estructuras son *XML*.

La ejecución de dicha librería consta solo de hacer las importaciones necesarias al principio de un archivo *Python*, dado a que BeautifulSoup fue pensado en la extracción de textos de páginas web, la cual lo hace una de las herramientas más utilizadas dentro de la creación de *Scrappers*, los cuales se encargan de extraer información (texto o archivos completos de un servidor) de páginas web con la intención de preservar la información almacenada dentro de ellas [34].

3.7.4. Scikit-learn

Es un módulo creado en *Python* para aprendizaje automático que cuenta con varios algoritmos de clasificación, regresión y agrupamiento que están diseñados para operar con las bibliotecas numéricas y científicas de Python conocidas como *NumPy* y *SciPy*. La variedad de algoritmos y utilidades que presenta lo hace una herramienta básica para programar y estructurar sistemas de análisis de datos y modelado estadístico [25].

Capítulo 4

Propuesta de solución

El desarrollo está integrado de cuatro etapas: Análisis de datos, pre-procesamiento, clasificación y evaluación. Las cuales deben realizarse por cada conjunto de datos utilizado (India, China y México).

La **Figura 4.1** muestra las etapas de la solución. Donde los datos de entrada son revisados para que el conjunto de datos a trabajar esté presentado sobre una estructura (tablas, registros o bases de datos) y realizar una clasificación directa. También se realiza un pre-procesamiento de datos sobre el mismo conjunto que al terminar realizará otra clasificación. Finalmente, se lleva a cabo una reducción de términos para generar nuevos conjuntos de datos con algoritmos de reducción de términos y evaluar cada conjunto de datos creado.

4.1. Recopilación y análisis de datos

Los datos abarcados en este trabajo son tomados de diferentes fuentes, como la investigación realizada por Chen *et al.* [35], el cual nos presenta una amplia gama de datos antropométricos estructurados de ciudadanos residentes de China de los años 2010 a 2016, con el propósito de remarcar una relación entre el índice de masa corporal (IMC) y la DMT2 con la edad de los pacientes. Este conjunto de datos presenta varios campos el cual nos permite saber si el registro (paciente) cuenta con un familiar que padeció de diabetes, diagnóstico a la enfermedad DMT2, valores antropométricos y

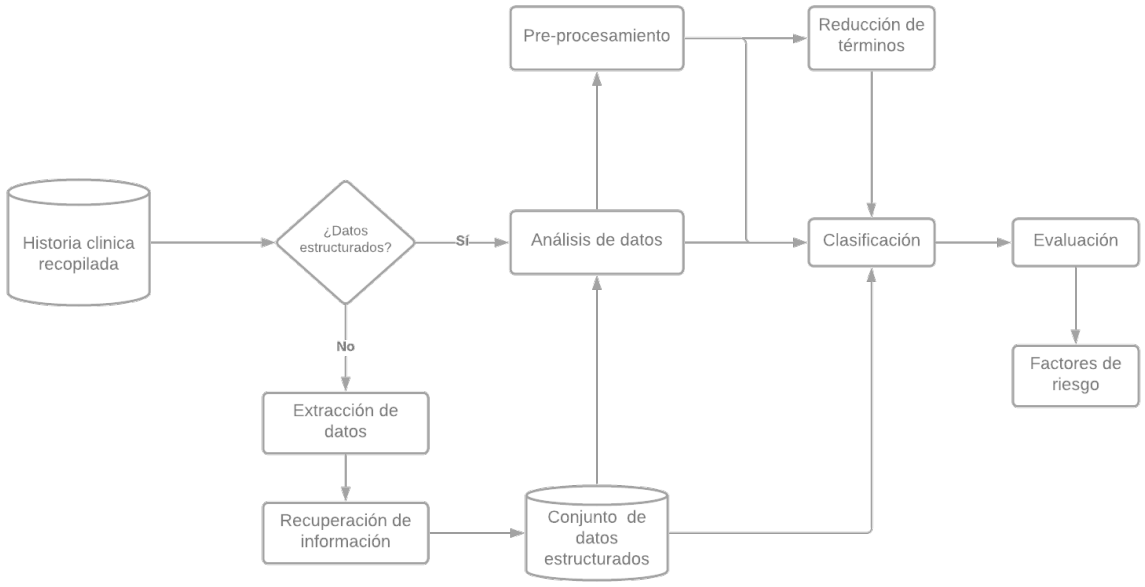


Figura 4.1: Diagrama de flujo.

ciertos estudios médicos. Pero ciertos campos se encuentran con resultados no válidos (sin datos o nulos) entre ellos datos como mediciones de colesterol *HDL*, *LDL*, glucosa y otros.

Otra fuente de datos utilizada en este tema es conocida como *Pima Indians Diabetes Database*, la cual es muy frecuentada en el estado del arte, ya que es un recurso de 768 mujeres con los atributos de cantidad de embarazos, edad, peso, IMC, presión arterial y un campo donde se reporta el resultado al pronóstico de DM [36].

Para finalizar con las fuentes de datos, tenemos los registros del sector salud ubicado en Carmen Xhan, cerca de la frontera México-Guatemala, el cual por el momento nos ha proporcionado datos de personas diabéticas, con atributos como la edad, IMC, presión arterial, índices de colesterol *HDL*, *LDL* y glucosa [37]. Desafortunadamente este conjunto de datos presenta el problema de no proporcionar a personas no diabéticas o que no padecen la enfermedad DMT2, lo cual hace inservible la clasificación binaria. Esto es solventado utilizando los registros médicos del médico-cirujano Neri Salvador Cancino Hernández, que atiende a varias personas con y sin la enfermedad DMT2. Para así lograr aumentar el conjunto de datos proveniente de México al combinar los conjuntos de datos.

En la **Figura 4.2** extiende la explicación del funcionamiento de la extracción de datos y recuperación de información de la **Figura 4.1**. El proceso se aplica cuando la historia clínica recuperada no se encuentre en una estructura (texto plano, imágenes o audios). El primer paso consta de ubicar la carpeta donde se encuentra la información no estructurada, para posteriormente realizar una conversión y lograr extraer texto legible. Posteriormente se aplica la normalización para el texto y la creación de un índice invertido para hacer la recuperación de información más rápida y terminar con un concentrado de datos estructurados (consultar **Anexos**).

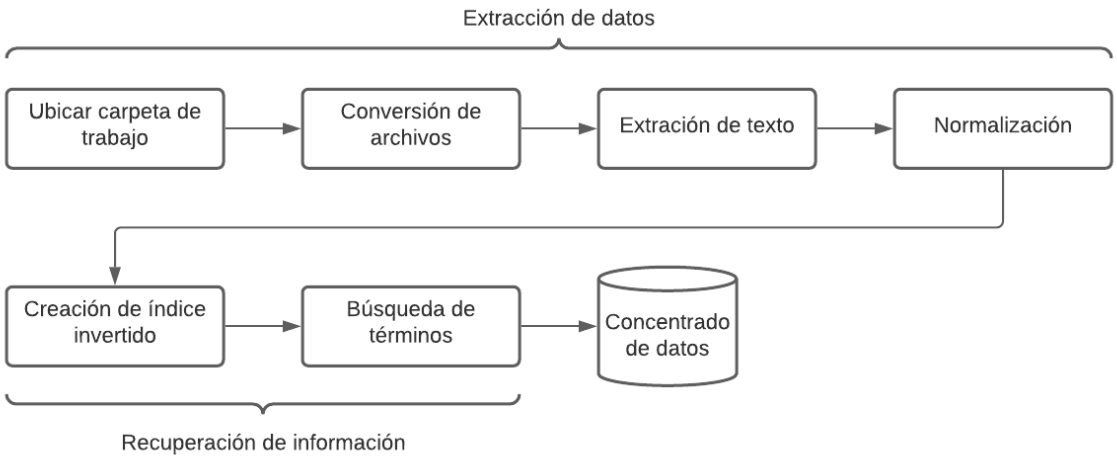


Figura 4.2: Extracción de datos y recuperación de información.

Para el análisis de datos se trabajó con cada conjunto de datos presentado por separado, utilizando la correlación de datos con ayuda de la herramienta *numpy* en donde podemos generar una matriz de correlación (basada en regresión lineal) con la intención de identificar la dispersión de los datos y ver que atributos tienen una mayor importancia dentro del conjunto de datos. Para que al realizar una reducción de términos se evite eliminar los atributos señalados por este análisis.

4.2. Pre-Procesado

El pre-procesamiento de los datos en general es la sustitución, eliminación o agrupamiento de los datos antes de aplicar cualquier algoritmo de clasificación. Un pre-procesado adecuado es de suma importancia porque la información dentro de los conjuntos es controlada de manera flexible lo que conlleva a errores de captura, resultados anormales o datos que faltan, siendo un problema a solucionar. Para solventar este

problema una propuesta más recurrida es eliminar los registros que no estén completos, pero esto provoca la omisión de muchos datos y no proporcionaría una buena clasificación. Otra propuesta es la sustitución por media, el cual funciona analizando los datos existentes completos para utilizar la media de cada atributo para completar los registros y así tener una base de conocimiento más completa, lo cual se traduce a un mejor pre-procesamiento.

En la **Figura 4.3** se muestra en un diagrama de flujo el funcionamiento de la sustitución por media, la cual empieza con la selección de un atributo y recorre cada uno de los datos (filas) del conjunto. Posteriormente revisa el resultado del etiquetado en donde si es un valor nulo este queda etiquetado para ser remplazado y en el caso de contener información (número) es sumado y contabilizado para sacar la media de los datos. Al terminar de recorrer todos los datos se remplazan los elementos etiquetados por la media calculada.

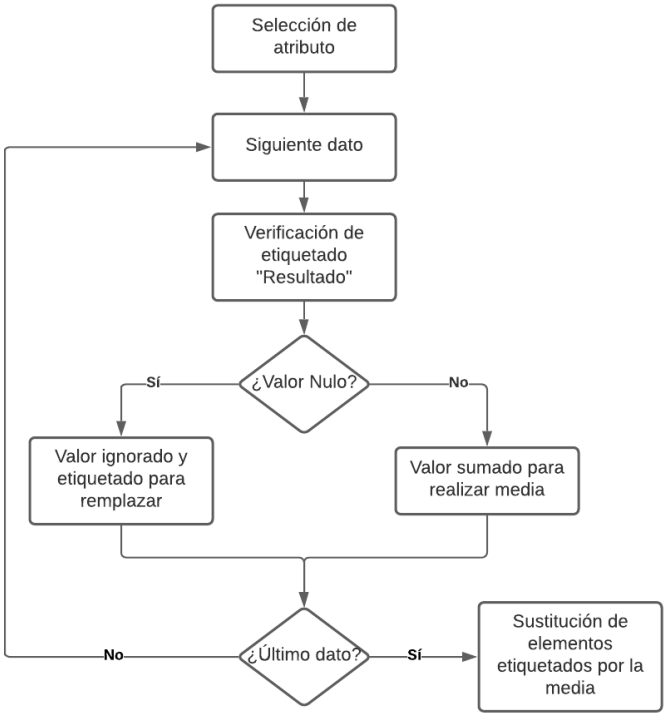


Figura 4.3: Procedimiento para sustitución por media.

Para llevar este proceso se hace uso del lenguaje de programación *Python* el cual nos ofrece varias herramientas para manipular grandes cantidades de datos de manera

eficiente a partir de archivos separados por comas (*csv*) y guardar los nuevos registros en un formato igual para el uso futuro [32].

Para utilizar los algoritmos de reducción de términos se hace uso de los conjuntos de datos resultantes del este apartado y eliminar atributos de cada conjunto de datos evitando retirar los marcados por el análisis de correlación y así generar nuevos conjuntos de datos con menos atributos. El propósito de esta reducción de datos sin alterar a los atributos más relacionados es la identificación de factores de riesgo del conjunto de datos, ya que los datos eliminados no presentaban relevancia y no estaban relacionados con algún otro atributo.

La **Tabla 4.1** muestra la estructura que debe de cumplir cada conjunto de datos para realizar los algoritmos de reducción de términos como PCA y *Non-Negative Matrix Factorization* (NMF) utilizando la librería de Scikit-learn dentro de un ambiente de *Python*. La primera columna es un identificador de la fila y la última columna es la etiqueta clasificadora de la fila, donde cero representa a una persona no diabética y uno representa una persona con DMT2. Por otra parte, las columnas que se encuentran en medio son los datos del conjunto.

Tabla 4.1: Estructura del conjunto de datos.
Los puntos en la tabla representan más datos.

I	**Embarazos**	**Glucosa**	**Edad**	**Presión**	**DPF**	...	**Outcome**
0	6	148	72	0.627	50	...	1
1	1	85	66	0.351	31	...	0
2	8	183	64	0.672	32	...	1
3	1	89	66	0.167	21	...	0
4	0	137	40	2.288	33	...	1
...	...	...	...	...	...	...	...

Al aplicar los algoritmos de reducción de términos nos dan una ponderación por cada atributo del conjunto de datos. El algoritmo PCA calcula valores de los auto-vectores que generalmente se presentan en una gráfica de barras la cual representa la importancia de cada atributo. Mientras que el algoritmo NMF cosiste en un vector con las puntuaciones de cada atributo.

Los resultados del algoritmo PCA consiste en calcular los valores de los auto-vectores, los cuales son usados posteriormente para presentar una gráfica de barras donde se representa la importancia de cada atributo con una ponderación de porcentajes. Mientras que los resultados de NMF consta de un vector de números en donde las puntuaciones más altas representan los atributos más representativos del conjunto de datos.

4.3. Clasificación

Para la selección de algoritmos de clasificación se optaron por los algoritmos usados dentro del estado del arte enfocados en la minería de datos (SMO, *TreeJ48*, *Random Forest* y *Naïve Bayes*) con la herramienta *WEKA* [38] utilizando el método de *Cross-Validation* (*CV*) para realizar las validaciones y obtener la matriz de confusión. Por aparte se realiza la implementación del algoritmo *Naïve Bayes* en el lenguaje de programación *Python* basado en el identificador ingenuo de Bayes de la librería de *WEKA*, donde arroja las métricas y matriz de confusión respectivos para la evaluación.

Cada clasificación se llevará a cabo por cada conjunto de datos, esto incluye los conjuntos de datos generados por el pre-procesamiento y la reducción de términos, lo cual nos da un total de tres ejecuciones del clasificador seleccionado por cada conjunto de datos utilizado.

4.4. Evaluación de algoritmos

Para la evaluación de los algoritmos de clasificación es un requerimiento tener presente la matriz de confusión (**Figura 4.4**) para realizar las operaciones necesarias y obtener cuatro métricas de evaluación. Si la matriz de confusión no está presente, se puede obtener un aproximado de las métricas utilizando la cantidad de registros correctamente clasificados contra el total de registros.

		Prediccion	
		0	1
Realidad	0	VN	FP
	1	FN	VP

Figura 4.4: Representación de matriz de confusión.

Las métricas que hacen uso de esta matriz nos comparan la predicción de los registros con el resultado de la predicción del algoritmo, las cuales son [39]:

- **Precisión:** Es el número de casos relevantes recuperados entre el número de casos recuperados.

- **Exhaustividad:** Expresa la proporción de casos relevantes recuperados, comparado con el total de los casos que son relevantes existentes, con total independencia de que estos, se recuperen o no.
- **Exactitud:** Mide el porcentaje de casos que el modelo ha acertado o clasificado correctamente.
- F_1**:** Es utilizado para combinar las medidas ***precisión*** y ***exhaustividad*** en un solo valor.

$$Precision = \frac{VP}{VP + FP} \tag{4.1}$$

$$Exhaustividad = \frac{VP}{VP + FN} \tag{4.2}$$

$$Exactitud = \frac{VP + VN}{VP + VN + FP + FN} \tag{4.3}$$

$$F_1 = 2 * \frac{Precision * Exhaustividad}{Precision + Exhaustividad} \tag{4.4}$$

Donde:

- Verdaderos positivos (VP): Son resultados positivos clasificados correctamente.
- Verdaderos negativos (VN): Son resultados negativos clasificados como positivos.
- Falsos positivos (FP): Son resultados negativos clasificados como positivos.
- Falsos negativos (FN): Son resultados negativos clasificados correctamente.

El objetivo de la evaluación es de medir la eficiencia del modelo creado en registros nuevos. Esta eficiencia es medida en porcentajes que pueden variar dependiendo del conjunto de datos utilizado.

Para determinar los atributos conocidos como factores de riesgo es necesaria la eliminación de atributos no relevantes para la clasificación, donde se recomienda utilizar métodos de reducción de términos ya que ofrecen una puntuación de los atributos utilizando probabilidad y estadística [40]. Al mejorar las métricas de evaluación de un conjunto de datos con menos atributos, se interpreta que los atributos presentes son los factores de riesgo.

Capítulo 5

Resultados

Para presentar los resultados de manera más ordenada, se hace uso de un apartado para explicar el contenido de cada conjunto de datos para posteriormente mostrar los resultados del pre-procesado junto con los atributos seleccionados por algoritmos de reducción de términos y terminar con la evaluación de los modelos generados por los clasificadores.

5.1. Conjuntos de datos

En este apartado se explicará el contenido referente a cada conjunto de datos utilizado junto con una breve explicación de cada uno y los motivos de la creación del conjunto de datos.

5.1.1. Primer conjunto de datos (PIDD)

El primer conjunto de datos es muy utilizado en el estado del arte ya que está basado en mediciones médicas proveniente del instituto nacional de diabetes y enfermedades digestivas y renales. Donde el objetivo es el predecir si un paciente padece de la enfermedad diabetes o no basándose en determinadas medidas de un diagnóstico. En particular, todos los pacientes en este conjunto de datos son mujeres de al menos 21

años y de ascendencia india, con un total de 500 registros no diabéticos y 268 diabéticos, también conocida como *Pima Indians Diabetes Dataset* (PIDD) [36]. En la **Tabla 5.1** se presentan los atributos de este conjunto de datos. La primera columna es el nombre de atributos utilizados, la segunda columna es una breve descripción del atributo y la última columna los valores mínimos y máximos dentro de cada atributo del conjunto de datos.

Tabla 5.1: Descripción del conjunto de datos PIDD [36]

Atributo	**Descripción**	**Min-Max**
Embarazos	Cantidad de embarazos.	0-17
Glucosa	Concentración de glucosa en plasma a dos horas en una prueba de tolerancia a la glucosa oral.	0-199
Presión sanguínea	Presión diastólica (mm Hg).	0-122
Grosor de la piel	Espesor del pliegue cutáneo del tríceps (mm).	0-99
Insulina	Insulina sérica de 2 horas (mu U / ml).	0-846
IMC	Índice de Masa Corporal (Kg/m^2).	0-67.1
PedigreeFunction	Función de árbol genealógico de la diabetes.	0.08-2.42
Edad	Edad en años.	21-81
Resultado	Resultado del paciente a la enfermedad de diabetes.	0-1

5.1.2. Segundo conjunto de datos (China)

El segundo conjunto de datos utilizado en esta investigación es realizado por los autores del artículo [35]. En donde se intenta identificar una relación entre la edad junto con el índice de masa corporal (IMC) y la diabetes en general en la población de China. El conjunto de datos es de diferentes ciudades de China (Shanghai, Beijing, Nanjing, Suzhou, entre otras) con pacientes mayores de 20 años con un total de 4,174 diabéticos de 211,835 pacientes registrados. En la **Tabla 5.2** se describen los atributos de este segundo conjunto de datos.

5.1.3. Tercer conjunto de datos (México)

El tercer y último conjunto de datos fue solicitado al sector salud ubicado en Carmen Xhan en la frontera de México-Guatemala (**Tabla 5.3**), en donde se solicitó las medidas de ciertos atributos de personas con padecimiento de la enfermedad DMT2, para

Tabla 5.2: Descripción del conjunto de datos China [35].

Atributo	**Descripción**	**Min-Max**
Edad	Edad en años.	20-99
IMC	Índice de Masa Corporal (Kg/m^2).	0-846
CC	Circunferencia de la cintura en centímetros.	13-1116.6
Presión sistólica	Presión sistólica (mm).	59-222
Presión diastólica	Presión diastólica (Hg).	38-164
HbA1c	Prueba de hemoglobina glicosilada.	0.59-6.99
Colesterol Total	Prueba de lipoproteínas.	0.02-17.84
Colesterol LDL	Lipoproteínas de baja densidad.	0-10.07
Colesterol HDL	Lipoproteínas de alta densidad.	0-10.4
Triglicéridos	Prueba de triglicéridos	0-32.64
Resultado	Resultado del paciente a la enfermedad de diabetes.	0-1

mantener la confidencialidad, la información sensible (nombre, residencia o número de teléfono) no fue solicitada. El conjunto de datos fue complementado con la ayuda del médico-cirujano Dr. Neri Salvador Cancino Hernández, el cual nos proporcionó atributos idénticos a los del sector salud junto con personas que no padecían de la enfermedad DMT2 aumentando la cantidad de registros total a 623 pacientes mexicanos mayores a 21 años en donde solamente 87 personas padecen DMT2 y 536 personas no padecen la enfermedad.

Tabla 5.3: Descripción del conjunto de datos México.

Atributo	**Descripción**	**Min-Max**
Edad	Edad en años.	21-81
Peso	Peso del paciente en Kilogramos.	33.5-120
IMC	Índice de Masa Corporal (Kg/m^2).	15.24-41.5
Presión sistólica	Presión sistólica (mm).	70-180
Presión diastólica	Presión diastólica (Hg).	40-113
Colesterol Total	Prueba de lipoproteínas.	0-299
Colesterol LDL	Lipoproteínas de baja densidad.	0-133
Colesterol HDL	Lipoproteínas de alta densidad.	0-51
Triglicéridos	Prueba de triglicéridos	0-1982
Resultado	Resultado del paciente a la enfermedad de diabetes.	0-1

5.2. Resultados del pre-procesado

En este apartado mostraremos el tratamiento de cada conjunto de datos para el manejo de datos que faltan y la generación de variantes del conjunto de datos al utilizar algoritmos de reducción de términos.

5.2.1. Primer conjunto de datos (PIID)

Al realizar el pre-procesado de datos en el primer conjunto de datos se detectaron 763 campos con valores no validos distribuidos entre los diferentes atributos, en donde el atributo con menor cantidad de datos que faltan fue glucosa, mientras que la insulina representaba casi a la mitad de los datos que faltan.

Al terminar con el pre-procesamiento de los datos se utiliza la herramienta *numpy* de *Python* para generar la matriz de correlación de los atributos para ver las relaciones más fuertes del conjunto de datos (**Figura 5.1**), en donde son presentadas en el espectro amarillo-blanco. Posteriormente se aplican algoritmos de reducción de términos para generar nuevos conjuntos de datos con una menor cantidad de atributos. Las relaciones más fuertes encontradas en este conjunto de datos recaen en el espectro amarillo-blanco donde los atributos glucosa-insulina (**Figura 5.2**) y grosor de la piel-IMC (**Figura 5.3**) se encuentran.

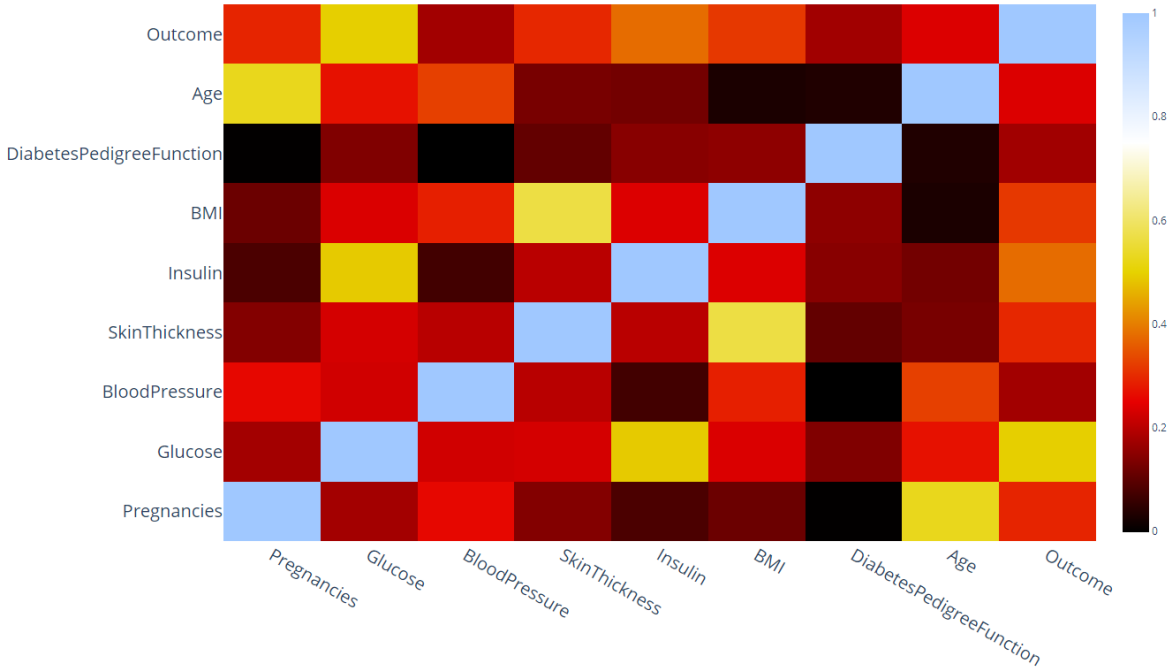


Figura 5.1: Matriz de correlación del conjunto de datos PIDD.

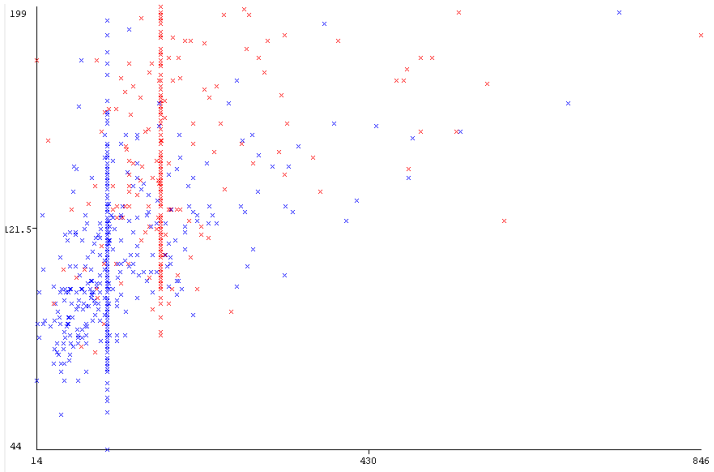


Figura 5.2: Correlación glucosa - insulina.

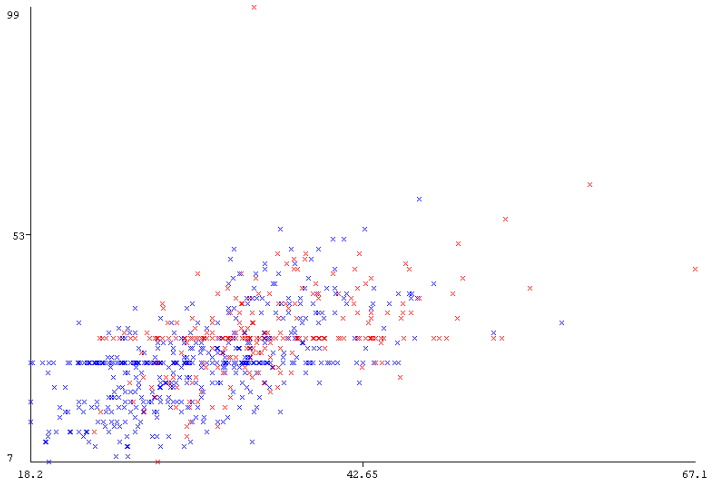


Figura 5.3: Correlación grosor de piel – IMC.

Una vez identificando los atributos con mayor relación se realizan los algoritmos de compresión de términos para determinar a los atributos serán eliminados. En la **Figura 5.4** se muestra la salida del algoritmo PCA, en donde los candidatos a ser eliminados son la presión y el grosor de la piel con resultados menores al 10 %.

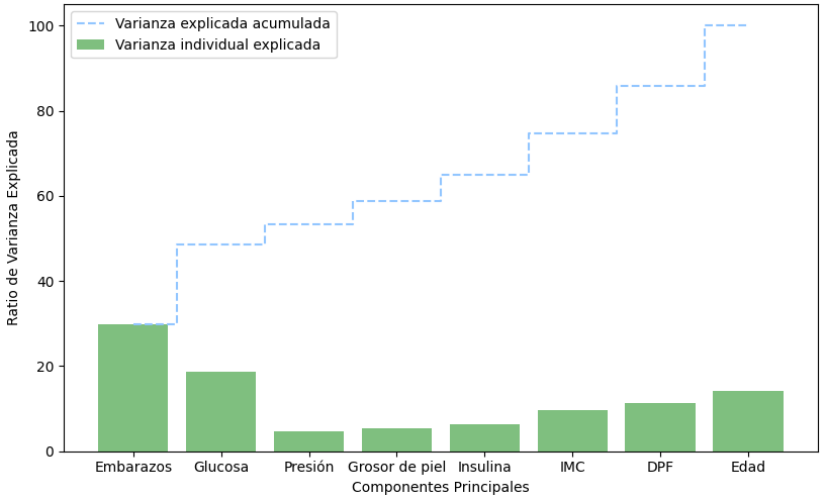


Figura 5.4: Representación gráfica de auto-valores con conjunto de datos PIDD.

Los resultados del algoritmo NMF presentados en la **Tabla 5.4** muestran que los atributos de embarazos y *PedigreeFunction* son los candidatos a ser eliminados por la baja puntuación obtenida.

Tabla 5.4: Resultados de NMF con conjunto de datos PIDD.

Atributo	**Puntuación**
Insulina	33.36660
Glucosa	24.76923
Presión sanguínea	14.03284
Edad	6.547377
IMC	6.363677
Grosor de piel	5.741375
Embarazos	0.762258
PedigreeFunction	0.094279

Por cada algoritmo de reducción de datos se crea un nuevo conjunto que es interpretado por la herramienta *WEKA*, el cual la mayoría de los casos es un archivo *csv*, obteniendo un total de cuatro archivos con cantidades diferentes de datos (**Tabla 5.5**). Para la eliminación de ciertos atributos se solicitó la opinión de un experto en el campo, para la revisión de los atributos eliminados, en las de este caso la eliminación de la presión sanguínea fue lo ideal, dado a que una baja cantidad de pacientes con diabetes padecen de hipertensión y si la padecen se puede tratar paralelamente [41].

Tabla 5.5: Conjuntos de datos generados por los algoritmos.

Todos los atributos	**Pre-procesado**	**Pre-procesado + PCA**	**Pre-procesado + NMF**
Embarazos	Embarazos	Embarazos	Glucosa
Glucosa	Glucosa	Glucosa	Presión sanguínea
Presión sanguínea	Presión sanguínea	Insulina	Grosor de la piel
Grosor de piel	Grosor de piel	IMC	Insulina
Insulina	Insulina	*PedigreeFunction*	IMC
IMC	IMC	Edad	Edad
PedigreeFunction	*PedigreeFunction*		
Edad	Edad		

5.2.2. Segundo conjunto de datos (China)

Para el conjunto de datos proveniente de China se aplica el mismo procesamiento que el primer conjunto de datos contando con la cantidad de 212,546 datos no válidos dentro del conjunto de datos. Donde la mayoría se encontraba en el colesterol *LDL* y *HDL* con 187,983 datos que faltaban entre ambos atributos. Mientras que la presión

arterial se encontraba con la cantidad mínima de datos no proporcionados (23 datos). La **Figura 5.5** muestra que las relaciones más fuertes se encuentran en el espectro azul-blanco, siendo el colesterol total-colesterol LDL y presión arterial sistólica-presión arterial diastólica las relaciones más fuertes del conjunto de datos.

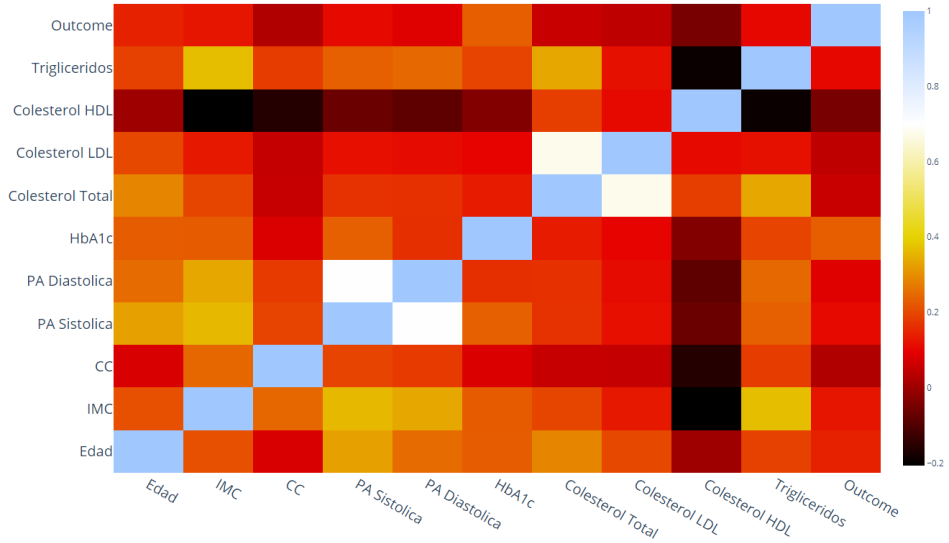


Figura 5.5: Matriz de correlación del conjunto de datos China.

En la **Figura 5.6** se muestra la relación del colesterol total con el colesterol *LDL* y es difícil apreciar la separación del conjunto dada a la gran cantidad de registros de personas no diabéticas. Sin embargo, se logra apreciar una ligera línea roja que indica una cantidad de personas que padecen de DMT2 en un rango que puede ser separado.

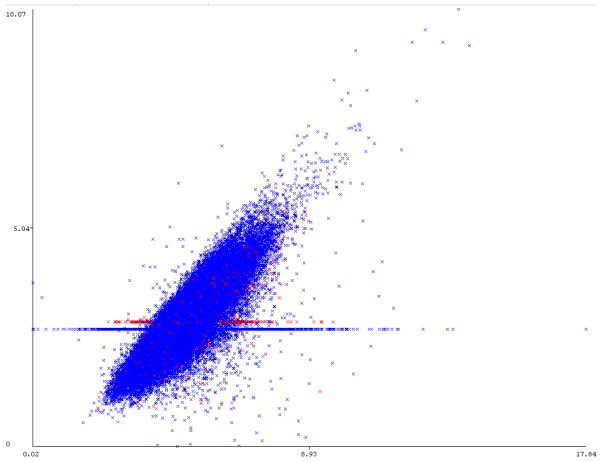


Figura 5.6: Relación colesterol total – colesterol *LDL*.

La **Figura 5.7** muestra los resultados del algoritmo PCA proveniente de los auto-vectores generados con la herramienta *Scikit-learn*. Donde los atributos candidatos a ser eliminados son: circunferencia de la cintura y la presión arterial sistólica por la baja puntuación obtenida (<=6 %) en comparación con otros atributos del conjunto de datos.

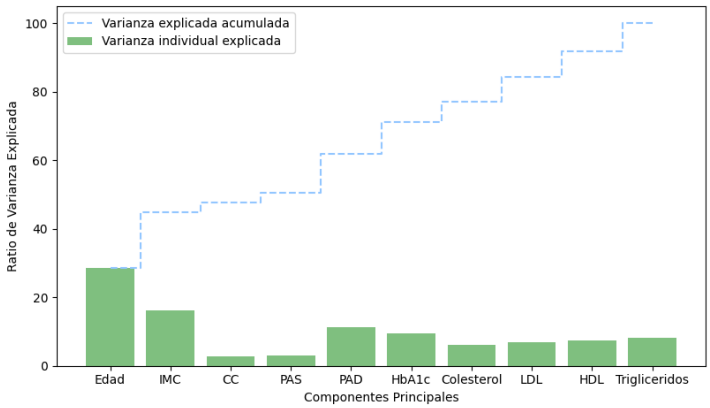


Figura 5.7: Representación gráfica de auto-valores con conjunto de datos China.

La **Tabla 5.6** representa el vector del algoritmo NMF que nos indica que atributos representan una menor cantidad de información (puntuaciones bajas) dentro del conjunto de datos se encuentran los candidatos: Triglicéridos, colesterol total *HDL*, *LDL* y HbA1c por su baja puntuación en comparación con los otros atributos.

Tabla 5.6: Resultados de NMF con conjunto de datos China.

Atributo	**Puntuación**
Presión sistólica	18.19267
Presión diastólica	11.31973
CC	10.68413
Edad	6.441724
IMC	3.524706
HbA1c	0.712762
Colesterol Total	0.711742
Colesterol LDL	0.413784
Triglicéridos	0.206820
Colesterol HDL	0.204603

En este particular caso el experto en el campo hizo un comentario respecto a la hemoglobina glucosilada (HbA1c), dado a que es una prueba que mide el promedio de glucosa en los últimos 3 meses. Es una prueba que se usa para diagnóstico y control,

por lo que se mantiene dentro de cada conjunto de datos. La **Tabla 5.7** muestra de manera resumida los atributos de cada conjunto particular creados en esta sección.

Tabla 5.7: Conjuntos de datos generados por los algoritmos.

Todos los atributos	**Pre-procesado**	**Pre-procesado + PCA**	**Pre-procesado + NMF**
Edad	Edad	Edad	Edad
IMC	IMC	IMC	IMC
CC	CC	Presión diastólica	CC
Presión sistólica	Presión sistólica	HbA1c	Presión sistólica
Presión diastólica	Presión diastólica	Colesterol total	Presión diastólica
HbA1c	HbA1c	Colesterol LDL	HbA1c
Colesterol total	Colesterol total	Colesterol HDL	Colesterol total
Colesterol LDL	Colesterol LDL	Triglicéridos	
Colesterol HDL	Colesterol HDL		
Triglicéridos	Triglicéridos		

5.2.3. Tercer conjunto de datos (México)

El último conjunto de datos proveniente del sector salud Carmen Xhan y registros del médico cirujano Dr. Neri Salvador, presento la menor cantidad de datos ausentes con un máximo de 678 datos no proporcionados con la mayor concentración de estos datos en el colesterol *HDL* y la menor cantidad en la prueba del colesterol total.

En la **Figura 5.8** muestra la matriz de correlación del conjunto de datos en donde las relaciones más fuertes se encuentran en el espectro amarillo-blanco, siendo IMC-peso y presión arterial sistólica-presión arterial diastólica las relaciones más fuertes del conjunto de datos.

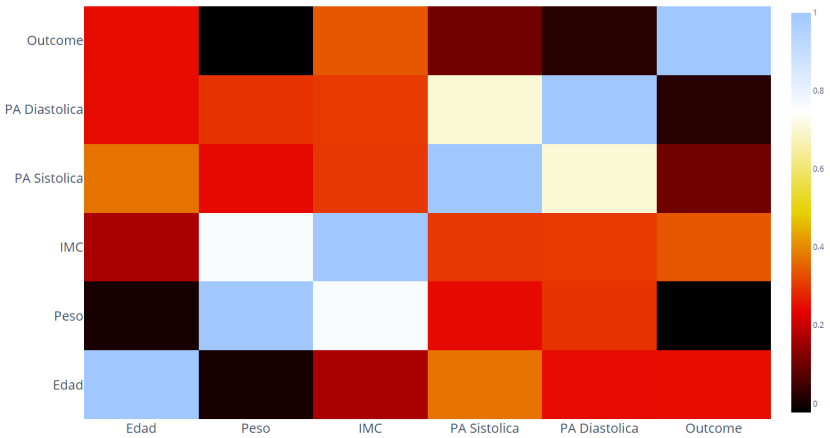


Figura 5.8: Matriz de correlación del conjunto de datos México.

La **Figura 5.9** muestra los resultados del algoritmo PCA, en donde se puede apreciar que los atributos de colesterol HDL, colesterol LDL y triglicéridos tienen los resultados más bajos (<=6 %), por lo que son los candidatos a ser eliminados.

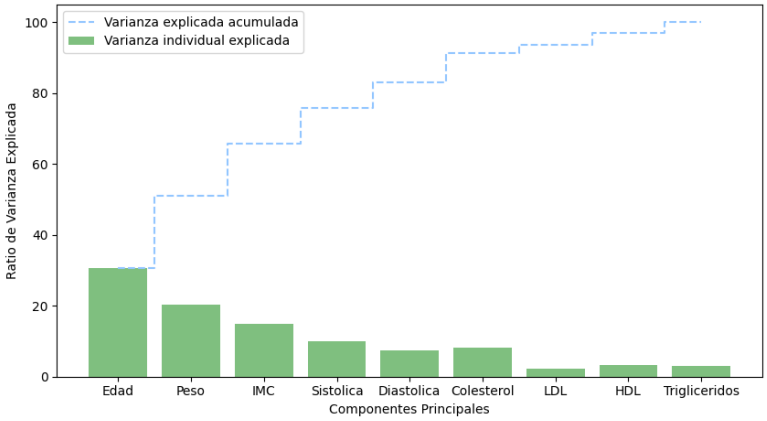


Figura 5.9: Representación de auto-valores con conjunto de datos México.

La **Tabla 5.8** muestra los resultados del algoritmo NMF que nos indica la puntuación obtenida por cada atributo dentro del conjunto de datos, donde los candidatos a eliminar son: la presión sistólica, triglicéridos y colesterol *HDL*.

Tabla 5.8: Resultados de NMF con conjunto de datos México.

Atributo	**Puntuación**
IMC	31.61724
Presión diastólica	20.22461
Colesterol LDL	16.88812
Colesterol Total	12.56167
Peso	11.00636
Edad de piel	9.656193
Colesterol HDL	7.304139
Triglicéridos	4.552653
Presión sistólica	4.035032

La **Tabla 5.9** muestra los conjuntos de datos resultantes de la sección de manera ordenada y simplificada.

Tabla 5.9: Conjuntos de datos resultantes de la sección.

Todos los atributos	**Pre-procesado**	**Pre-procesado + PCA**	**Pre-procesado + NMF**
Edad	Edad	Edad	Edad
Peso	Peso	Peso	Peso
IMC	IMC	IMC	IMC
presión sistólica	presión sistólica	presión sistólica	presión diastólica
presión diastólica	presión diastólica	presión diastólica	Colesterol Total
Colesterol total	Colesterol total	Colesterol total	Colesterol *LDL*
Colesterol *LDL*	Colesterol *LDL*		
Colesterol *HDL*	Colesterol *HDL*		
Triglicéridos	Triglicéridos		

5.3. Clasificación y evaluación

En esta sección se presentan los resultados experimentales de los algoritmos de clasificación con los conjuntos de datos. Para los experimentos se utiliza los algoritmos de clasificación implementados en la herramienta *WEKA*.

5.3.1. Primer conjunto de datos (PIDD

A continuación, se presentan los resultados experimentales de tres algoritmos de clasificación: *TreeJ48*, *Naïve Bayes* y SMO. En la **Tabla 5.10** se presentan los resultados del algoritmo de clasificación *TreeJ48*. Como puede observarse se logra una precisión de 86.5 %, exhaustividad de 87.2 % y exactitud de con pre-procesamiento de datos y PCA. Mientras que el trabajo [10] y [11], que nos presentaron un algoritmo similar obtienen porcentajes de 66.5 % y 78.17 % de exactitud respectivamente.

Tabla 5.10: Resultados del conjunto de datos PIDD usando *TreeJ48*.

Tratamiento	*Fuente*	*Precisión*	*Exhaustividad*	*Exactitud*	F_1
Atributos completos	[10]	N/R	N/R	66.5%	N/R
Limpieza de datos faltantes	[11]	N/R	N/R	78.17%	N/R
N/R	[12]	72.15%	58.96%	77.14%	64.89%
N/R	[13]	N/R	N/R	78.5%	N/R
N/R	[14]	78.60%	76.50%	N/R	77.10%
Pre-procesado	Propia	82.4%	78.7%	86.7%	80.5%
Pre-procesado + PCA	Propia	**86.5%**	**85.0%**	**87.2%**	**80.9%**
Pre-procesado + NMF	Propia	86.1%	86.2%	86.0%	86.1%

***N/R:** No reporta.

Para el caso del algoritmo de *Naïve Bayes* (**Tabla 5.11**) se obtienen resultados de 76.9 % en precisión y 77.0 % de exactitud con un conjunto de datos pre-procesado, lo cual es una gran mejora en comparación con el trabajo mostrado por los autores del trabajo [12] a pesar de utilizar el mismo método de evaluación (CV). Así mismo las métricas reportadas por los autores del trabajo [14] se obtiene un aumento bajo (0.2-0.3 %) al realizar un pre-procesado de los datos junto con una eliminación de atributos mediante algoritmos de reducción de términos (PCA). Sin embargo, no es suficiente para superar la métrica reportadas por el trabajo [13] utilizando la misma evaluación del algoritmo. Desafortunadamente dicho trabajo no muestra todas las métricas de evaluación, pre-procesado o matriz de confusión utilizada para llegar al resultado demostrado con el conjunto de datos.

La **Tabla 5.12** muestra el algoritmo de clasificación SMO donde los mejores resultados son los del pre-procesado alcanzando porcentajes del 79.9 % en exactitud y 78.3 % de F1, superando al trabajo [12]. Mientras que los resultados de los conjuntos de da-

Tabla 5.11: Resultados del conjunto de datos PIDD usando *Naïve Bayes*.

Tratamiento	*Fuente*	*Precisión*	*Exhaustividad*	*Exactitud*	F_1
N/R	[12]	66.80%	60.07%	75.65%	63.26%
N/R	[13]	N/R	N/R	78.1%	N/R
N/R	[14]	76.7%	77.00%	N/R	76.8%
Atributos completos	Propia	75.9%	76.3%	76.3%	76.0%
Pre-procesado	Propia	76.9%	77.1%	77.0%	77.0%
Pre-procesado + PCA	Propia	**77.0%**	**77.5%**	**77.4%**	**77.0%**
Pre-procesado + NMF	Propia	76.2%	76.7%	76.6%	76.3%

***N/R:** No reporta.

tos con menos atributos bajan su precisión un 0.8-0.9 % en comparación con el mejor resultado de la tabla.

Tabla 5.12: Resultados del conjunto de datos PIDD usando SMO.

Tratamiento	*Fuente*	*Precisión*	*Exhaustividad*	*Exactitud*	F_1
N/R	[12]	77.33%	53.36%	76.95%	61.77%
Atributos completos	Propia	77.3%	77.3%	77.3%	76.3%
Pre-procesado	Propia	**78.5%**	**78.9%**	**79.9%**	**78.3%**
Pre-procesado + PCA	Propia	77.7%	78.1%	78.1%	77.4%
Pre-procesado + NMF	Propia	77.4%	77.9%	77.8%	77.1%

***N/R:** No reporta.

5.3.2. Segundo conjunto de datos (China)

A continuación, se presentan los resultados experimentales de tres algoritmos de clasificación: *TreeJ48*, *Naïve Bayes* y *Random Forest*. En la **Tabla 5.13** se presentan los resultados del algoritmo *TreeJ48* donde se encuentra la puntuación más alta, ya que las métricas se encuentran por arriba del 95 %. Al realizar un pre-procesado y reducción mediante PCA, se puede observar que los resultados mejoran significativamente, pero quedan muy cerca de un resultado perfecto.

Tabla 5.13: Resultados del conjunto de datos China usando *TreeJ48*.

Tratamiento	*Precisión*	*Exhaustividad*	*Exactitud*	F_1
Atributos completos	97.1%	98.0%	98.0%	97.2%
Pre-procesado	98.5%	98.6%	**99.8%**	98.4%
Pre-procesado + PCA	**98.5%**	**98.7%**	98.6%	**98.5%**
Pre-procesado + NMF	97.0%	98.0%	98.0%	97.2%

El algoritmo de *Naïve Bayes* (**Tabla 5.14**) presenta una constante en los resultados de precisión al comparar la implementación de un pre-procesado y una reducción de términos al conjunto de datos. La constante en la precisión puede ser por cantidad de datos que tiene cada conjunto de datos. Sin embargo, las otras métricas (Exhaustividad, Exactitud, F_1) muestran una pequeña mejoría al aplicar un pre-procesado y algoritmos de reducción de términos. Donde la mejor métrica registrada es F_1 con el conjunto reducido con el algoritmo NMF.

Tabla 5.14: Resultados del conjunto de datos de China usando *Naïve Bayes*.

Tratamiento	*Precisión*	*Exhaustividad*	*Exactitud*	F_1
Atributos completos	97.2%	93.7%	93.7%	95.3%
Pre-procesado	97.2%	93.5%	93.4%	95.1%
Pre-procesado + PCA	97.2%	94.5%	94.5%	95.7%
Pre-procesado + NMF	**97.2%**	**94.7%**	**94.7%**	**95.8%**

El último algoritmo de clasificación utilizado en este conjunto de datos es el algoritmo *Random Forest* (**Tabla 5.15**), el cual presenta resultados del 98.6 % en precisión y 98.7 % de exactitud al aplicar un pre-procesado y reducción de términos mediante el algoritmo PCA.

Tabla 5.15: Resultados del conjunto de datos China usando *Random Forest*.

Tratamiento	*Precisión*	*Exhaustividad*	*Exactitud*	F_1
Atributos completos	97.2%	98.0%	98.0%	97.2%
Pre-procesado	98.6%	98.7%	98.7%	98.5%
Pre-procesado + PCA	**98.6%**	**98.7%**	**98.7%**	**98.5%**
Pre-procesado + NMF	97.2%	98.0%	98.0%	97.3%

La **Tabla 5.16** y **Tabla 5.17** muestra la matriz de confusión resultante de los modelos creados con los atributos completos y el pre-procesado respectivamente. Donde se puede observar que la cantidad de registros positivos correctamente clasificados con el conjunto de datos pre-procesado (1623 registros) es superior a la del conjunto con todos los atributos (209 registros). Demostrando que al utilizar un pre-procesado ayuda a realizar una mejor clasificación.

Tabla 5.16: Matriz de confusión resultante con atributos completos.

		Predicción	
Realidad		***No diabético***	***Diabético***
	No diabético	207445	214
	Diabético	3965	**209**

Tabla 5.17: Matriz de confusión resultante con pre-procesado.

		Predicción	
Realidad		***No diabético***	***Diabético***
	No diabético	207461	198
	Diabético	2551	**1623**

5.3.3. Tercer conjunto de datos (México)

Finalmente, para el conjunto de datos de México se empieza con el uso del algoritmo *TreeJ48* (**Tabla 5.18**). En donde podemos observar que los mejores resultados obtenidos son de 90.0 % de precisión y 90.5 % de exactitud. Estos resultados son obtenidos con el conjunto pre-procesado con reducción de términos mediante el algoritmo NMF.

Tabla 5.18: Resultados del conjunto de datos México usando *TreeJ48*.

Tratamiento	*Precisión*	*Exhaustividad*	*Exactitud*	F_1
Atributos completos	88.1%	89.2%	89.2%	88.2%
Pre-procesado	89.7%	90.2%	90.2%	89.9%
Pre-procesado + PCA	89.7%	90.4%	90.3%	89.9%
Pre-procesado + NMF	**90.0%**	**90.5%**	**90.5%**	**90.2%**

Para el algoritmo de *Naïve Bayes* (**Tabla 5.19**) demuestra una clasificación de 86 % de exactitud sin un pre-procesamiento. Pero, al utilizar un pre-procesamiento en los datos podemos apreciar que las métricas bajan por 1 % en todas las métricas. Sin embargo, al retirar atributos del conjunto con PCA podemos observar una muy baja mejora (1 %) en las métricas de exhaustividad y exactitud que logran superar al conjunto de datos con atributos completos.

Tabla 5.19: Resultados del conjunto de datos México usando *Naïve Bayes*.

Tratamiento	*Precisión*	*Exhaustividad*	*Exactitud*	F_1
Atributos completos	**85.6%**	86.0%	86.0%	85.8%
Pre-procesado	84.3%	85.1%	85.0%	84.6%
Pre-procesado + PCA	85.5%	**86.7%**	**86.6%**	**86.0%**
Pre-procesado + NMF	85.1%	85.9%	85.8%	85.4%

La **Tabla 5.20** muestra los resultados el algoritmo *Random Forest*, el cuál demostró dar las mejores métricas del conjunto de datos alcanzando un 91.1 % de exactitud sin utilizar un pre-procesamiento o reducción de términos. Consecuentemente al realizar un pre-procesado las cuatro métricas aumentan considerablemente (3-4 %), pero se observa que al retirar atributos del conjunto de datos las métricas decaen un 1 % la cual sigue siendo una buena clasificación.

Tabla 5.20: Resultados del conjunto de datos México usando *Random Forest*.

Tratamiento	*Precisión*	*Exhaustividad*	*Exactitud*	F_1
Atributos completos	90.7%	91.3%	91.1%	90.5%
Pre-procesado	**94.5%**	**94.4%**	**94.3%**	**93.9%**
Pre-procesado + PCA	93.9%	93.9%	93.9%	93.3%
Pre-procesado + NMF	94.2%	94.2%	94.2%	93.7%

Finalmente, la **Tabla 5.21** muestra un recopilado de los mejores resultados por cada conjunto de datos utilizando las métricas Exactitud y F_1, incluyendo los atributos utilizados para alcanzar los resultados presentados y el algoritmo clasificador con los que se obtuvieron los resultados. Los atributos presentados en la **Tabla 5.21** son considerados los factores de riesgo relevantes, ya que con estos atributos se obtiene una clasificación robusta con resultados por encima del 80 % de exactitud al clasificar.

Tabla 5.21: Mejor resultado por cada conjunto de datos.

Conjunto de datos	*Exactitud*	F_1	*Atributos utilizados*	*Clasificador*
Primer conjunto (PIDD)	87.2%	80.9%	Embarazos, glucosa, insulina, IMC, *PedigreeFunction* y edad	*TreeJ48*
Segundo conjunto (China)	98.7%	98.5%	Edad, IMC, presión diastólica, HBA1c colesterol total, LDL, HDL y triglicéridos	*Random Forest*
Tercer conjunto (México)	94.2%	93.7%	Edad, Peso, IMC, presión diastólica, Colesterol Total y colesterol LDL	*Random Forest*

Capítulo 6

Conclusiones

El diagnóstico temprano de enfermedades crónicas es una de las problemáticas del campo médico que ha sido solventado con la ayuda del aprendizaje automático, proporcionando al médico una segunda opinión certera y comprobable. En este documento recopilamos información de diferentes fuentes para comparar varios algoritmos que se han usado para la tarea de clasificar la enfermedad DMT2 basado en diferentes conjuntos de datos provenientes de diferentes países, para la identificación correcta de pacientes que padecen esta enfermedad. Con la intención de encontrar los factores de riesgo al reducir la cantidad de atributos no relevantes utilizando algoritmos de reducción de términos.

La opinión y comentarios del médico experto dentro del campo son referentes a los atributos seleccionados, concordando que teniendo los campos presentados es posible realizar un diagnóstico de DMT2, al igual que la exactitud obtenida con el conjunto de datos de China (98.7 %) al eliminar los atributos marcados por el algoritmo PCA. Llegando a la conclusión de que se deberían usar los datos que proporcionen un mayor porcentaje de exactitud, lo que se puede traducir a mejores clasificaciones.

El conjunto de datos proveniente de la india muestra porcentajes de 80.9 % y 87.2 % en las métricas F_1 y exactitud respectivamente utilizando el algoritmo *TreeJ48*, con los atributos de: embarazos, glucosa, insulina, IMC, *PedigreeFunction* y edad. Mientras que para el conjunto de datos de China mostró porcentajes de 98.5 % y 98.7 % en F_1 y exactitud con el algoritmo *Random Forest* utilizando los atributos: edad, IMC, presión diastólica, HBA1c Colesterol total, colesterol *LDL*, colesterol *HDL* y Triglicéridos. Finalmente, el conjunto de datos proveniente de México alcanza porcentajes de 93.7 % y

94.2 % en F_1 y exactitud nuevamente con el algoritmo *Random Forest* y con los atributos: edad, peso, IMC, presión arterial diastólica, Colesterol Total y colesterol *LDL*. En donde los atributos en común entre los tres conjuntos son: el IMC y la edad, que son atributos que representan el sedentarismo de una persona.

Otro tema a discusión es el pre-procesamiento de los datos, ya que en este documento se utilizó un pre-procesado de sustitución por media, que ayuda a mejorar las métricas de evaluación en comparación de no hacer nada o de simplemente eliminar los registros incompletos. Una mejora podría ser la implementación de métodos más complejos para remplazar estos datos para evitar seccionar a ciertos atributos por falta de datos.

La reducción de dimensiones es una de las herramientas más poderosas con la que cuenta la minería de datos, ya que nos permite poder retirar datos sin perder información, como es demostrado en los resultados de este trabajo, ya que no se pierden grandes cantidades en las métricas presentadas al retirar más de un atributo a cualquier conjunto de datos. Gracias a esta reducción de datos, junto con análisis de correlación entre los atributos, podemos descartar la información no relevante dentro de los conjuntos de datos y así enfocarnos en la información relevante, que en este caso son los factores de riesgo que determinan en un paciente el padecimiento de la enfermedad DMT2.

Bibliografía

[1] D. Riaño, F. Real, J. A. López-Vallverdú, F. Campana, S. Ercolani, P. Mecocci, R. Annicchiarico y C. Caltagirone, "An ontology-based personalization of healthcare knowledge to support clinical decisions for chronically ill patients", *Journal of Biomedical Informatics*, vol. 45, n.º 3, págs. 429-446, 2012, ISSN: 15320464. DOI: 10.1016/j.jbi.2011.12.008.

[2] Yamilé, Miguel Soca, P. E. Sarmiento Teruel, A. Luis, Mariño Soler, Y. LLorente Columbié, T. Rodríguez Graña y M. Peña González, "Prevalencia de enfermedades crónicas no transmisibles y factores de riesgo en adultos mayores de Holguín", *Revista Finlay*, vol. 7, n.º 3, págs. 155-167, 2017. dirección: http://scielo.sld.cu/scielo.php?script=sci%7B%5C_%7Darttext%7B%5C&%7Dpid=S2221-24342017000300002%7B%5C&%7Dlng=en%7B%5C&%7Dtlng=en.

[3] N. S. Rodríguez-Rivera, P. Cuautle-Rodríguez, F. Castillo-Nájera y J. A. Molina-Guarneros, "Identification of genetic variants in pharmacogenetic genes associated with type 2 diabetes in a mexican-mestizo population", *Biomedical Reports*, vol. 7, n.º 1, págs. 21-28, 2017, ISSN: 20499442. DOI: 10.3892/br.2017.921.

[4] Dirección General de Información en Salud, ed., *Manual del Expediente Clínico Electrónico*. Secretaría de Salud, 2011.

[5] M. Teresa Romá-Ferri y M. Palomar, "Análisis de terminologías de salud para su utilización como ontologías computacionales en los sistemas de información clínicos", *Gaceta Sanitaria*, vol. 22, n.º 5, págs. 421-433, 2008, ISSN: 02139111. DOI: 10.1157/13126923. dirección: http://dx.doi.org/10.1157/13126923.

[6] J. A. O. Moreno, *El expediente clínico electrónico universal en México*, 2018.

[7] J. M. Candela, *¿Cuáles son los factores de riesgo para desarrollar diabetes mellitus tipo 2?*, 2015.

[8] Y. Llorente, P. Enrique, D. Rivas e Y. Borrego, "Factores de riesgo asociados con la aparición de diabetes mellitus tipo 2 en personas adultas", *Revista Cubana de Endocrinología*, vol. 27, n.º 2, págs. 123-133, 2016. dirección: http://scielo.sld.cu/scielo.php?script=sci%7B%5C_%7Darttext%7B%5C&%7Dpid=S1561-29532016000200002.

[9] K. Vembandasamy, R. Sasipriya y E. Deepa, "Heart Diseases Detection Using Naive Bayes Algorithm", *International Journal of Innovative Science, Engineering & Technology*, vol. 2, n.º 9, págs. 441-444, 2015.

[10] O. Chan, J. Peña, J. Vianne y M. Zapata, "Construcción De Un Modelo De Predicción Para Apoyo Al Diagnóstico De Diabetes (Construction of a Prediction Model To Support the Diabetes Diagnosis)", *Pistas Educativas*, vol. 40, n.º 130, págs. 2105-2122, 2018.

[11] A. A. AlJarullah, "Decision tree discovery for the diagnosis of type II diabetes", *2011 International Conference on Innovations in Information Technology, IIT 2011*, págs. 303-307, 2011. DOI: 10.1109/INNOVATIONS.2011.5893838.

[12] P. Hemant y T. Pushpavathi, "A novel approach to predict diabetes by Cascading Clustering and Classification", *2012 3rd International Conference on Computing, Communication and Networking Technologies, ICCCNT 2012*, 2012. DOI: 10.1109/ICCCNT.2012.6396069.

[13] V. V. V, "Prediction and Diagnosis of Diabetes Mellitus -A Machine Learning Approach", n.º December, págs. 122-127, 2015.

[14] S. Sa'di, A. Maleki, R. Hashemi, Z. Panbechi y K. Chalabi, "Comparison of Data Mining Algorithms in the Diagnosis of Type Ii Diabetes", *International Journal on Computational Science & Applications*, vol. 5, n.º 5, págs. 1-12, 2015. DOI: 10.5121/ijcsa.2015.5501.

[15] S. Perveen, M. Shahbaz, A. Guergachi y K. Keshavjee, "Performance Analysis of Data Mining Classification Techniques to Predict Diabetes", *Procedia Computer Science*, vol. 82, n.º March, págs. 115-121, 2016, ISSN: 18770509. DOI: 10.1016/j.procs.2016.04.016. dirección: http://dx.doi.org/10.1016/j.procs.2016.04.016.

[16] H. M. Deberneh e I. Kim, "Prediction of type 2 diabetes based on machine learning algorithm", *International Journal of Environmental Research and Public Health*, vol. 18, n.º 6, págs. 9-11, 2021, ISSN: 16604601. DOI: 10.3390/ijerph18063317.

[17] E. Denova-Gutiérrez, H. Lopez-Gatell, J. L. Alomia-Zegarra, R. López-Ridaura, C. A. Zaragoza-Jimenez, D. D. Dyer-Leal, R. Cortés-Alcala, T. Villa-Reyes, R. Gutiérrez-Vargas, K. Rodríguez-González, C. Escondrillas-Maya, T. Barrientos-Gutiérrez, J. A. Rivera y S. Barquera, "The association between obesity, type 2 diabetes, and hypertension with severe COVID-19 on admission among Mexicans", *Obesity*, vol. 00, n.º 00, págs. 1-7, 2020, ISSN: 1930739X. DOI: 10.1002/oby.22946.

[18] B. Cariou, S. Hadjadj, M. Wargny, M. Pichelin, A. Al-Salameh, I. Allix, C. Amadou, G. Arnault, F. Baudoux, B. Bauduceau, S. Borot, M. Bourgeon-Ghittori, O. Bourron, D. Boutoille, F. Cazenave-Roblot, C. Chaumeil, E. Cosson, S. Coudol, P. Darmon, E. Disse, A. Ducet-Boiffard, B. Gaborit, M. Joubert, V. Kerlan, B. Laviolle, L. Marchand, L. Meyer, L. Potier, G. Prevost, J. P. Riveline, R. Robert, P. J. Saulnier, A. Sultan, J. F. Thébaut, C. Thivolet, B. Tramunt, C. Vatier, R.

Roussel, J. F. Gautier y P. Gourdy, "Phenotypic characteristics and prognosis of inpatients with COVID-19 and diabetes: the CORONADO study", *Diabetologia*, vol. 63, n.º 8, págs. 1500-1515, 2020, ISSN: 14320428. DOI: 10.1007/s00125-020-05180-x.

[19] M. Nedyalkova, S. Madurga y V. Simeonov, "Combinatorial k-means clustering as a machine learning tool applied to diabetes mellitus type 2", *International Journal of Environmental Research and Public Health*, vol. 18, n.º 4, págs. 1-10, 2021, ISSN: 16604601. DOI: 10.3390/ijerph18041919.

[20] B. Cuji, W. Gavilanes y R. Sanchez, "Modelo predictivo de deserción estudiantil basado en arboles de decisión Predictive model of student dropout based on decision trees", *Espacios*, vol. 38, n.º 55, pág. 17, 2017. dirección: http://www.revistaespacios.com/a17v38n55/a17v38n55p17.pdf.

[21] J. García, J. M. Molina, A. Berlanga, M. A. Patricio, Á. L. Bustamante y W. R. Padilla, *Ciencia de datos. Técnicas analíticas y aprendizaje estadístico*. 2018, pág. 445, ISBN: 9788494731969. dirección: https://d1wqtxts1xzle7.cloudfront.net/64031156/Ciencia%7B%5C_%7Dde%7B%5C_%7Ddatos%7B%5C_%7D2018.pdf?1595866723=%7B%5C&%7Dresponse-content-disposition=inline%7B%5C%%7D3B+filename%7B%5C%7D3DCiencia%7B%5C_%7Dde%7B%5C_%7Ddatos%7B%5C_%7DTecnicas%7B%5C_%7Danaliticas%7B%5C_%7Dy%7B%5C_%7Da.pdf%7B%5C&%7DExpires=1599881630%7B%5C&%7DSignature=adAexnneE8eXc9VlTrWn-uKxVT1GlJvP88QgXG1vuwXZ9l.

[22] H. Kumar, *Scaling vs Normalization*, 2018. dirección: https://kharshit.github.io/blog/2018/03/23/scaling-vs-normalization%7B%5C#%7D:%7B~%7D:text=5%20mins%20read,while%20using%20machine%20learning%20algorithms. (visitado 20-11-2020).

[23] D. L. Poole, A. Mackworth y R. G. Goebel, "Computational Intelligence and Knowledge", *Computational Intelligence: A Logical Approach*, n.º Ci, págs. 1-22, 1998. dirección: https://www.cs.ubc.ca/%7B~%7Dpoole/ci.html.

[24] V. Torra, *La inteligencia artificial*, 2011. dirección: http://www.fgcsic.es/lychnos/es%7B%5C_%7Des/articulos/inteligencia%7B%5C_%7Dartificial.

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot y E. Duchesnay, "Scikit-learn: Machine Learning in {P}ython", *Journal of Machine Learning Research*, vol. 12, págs. 2825-2830, 2011.

[26] R. Barrientos, N. Cruz, H. Acosta, I. Rabatte, M. Gogeascoechea, P. Pavón y S. Blázquez, "Árboles De Decisión Como Herramienta En El Diagnóstico Médico", *Articulo Original*, págs. 20-24, 2009. dirección: https://www.uv.mx/rm/num%7B%5C_%7Danteriores/revmedica%7B%5C_%7Dvol9%7B%5C_%7Dnum2/articulos/arboles.pdf.

[27] J. R. Amat, *Máquinas de Vector Soporte (SVM) con Python*, 2020. dirección: https://www.cienciadedatos.net/documentos/py24-svm-python.html (visitado 04-04-2021).

[28] R. Elshawi y S. Sakr, "Big data systems meet machine learning challenges: Towards big data science as a service", *arXiv*, 2017.

[29] O. Rodríguez Hernández, *Temas de análisis estadístico multivariado*, Primera Ed, G. Carazo G., ed. Costa rica: Universidad de Costa Rica, 1998, pág. 169, ISBN: 9977-67-490-6. dirección: https://books.google.com.mx/books?id=g-IT184TSS4C%7B%5C&%7Dprintsec=frontcover%7B%5C#%7Dv=onepage%7B%5C&%7Dq%7B%5C&%7Df=false.

[30] R. A. FISHER, "THE USE OF MULTIPLE MEASUREMENTS IN TAXONOMIC PROBLEMS", *Annals of Eugenics*, vol. 7, n.º 2, págs. 179-188, sep. de 1936, ISSN: 20501420. DOI: 10.1111/j.1469-1809.1936.tb02137.x. dirección: http://doi.wiley.com/10.1111/j.1469-1809.1936.tb02137.x.

[31] D. Solis, *Ejemplo de Reducción de Dimensionalidad con la Base de Datos Iris.* dirección: http://rstudio-pubs-static.s3.amazonaws.com/159299%7B%5C_%7Dd808bde45c9c4e67bdbde58724c4b2d6.html%20https://www.coursehero.com/file/66537314/LIBRO-U3-EjemploReduccion-BD-IRISpdf/.

[32] R. Wade, "Reading CSV Files", en *Advanced Analytics in Power BI with R and Python: Ingesting, Transforming, Visualizing.* Berkeley, CA: Apress, 2020, págs. 151-175, ISBN: 978-1-4842-5829-3. DOI: 10.1007/978-1-4842-5829-3_3. dirección: https://doi.org/10.1007/978-1-4842-5829-3%7B%5C_%7D3.

[33] Scanny, *Pydocx*, 2019. dirección: https://pypi.org/project/python-docx/ (visitado 07-08-2020).

[34] R. Mitchell, *Web Scraping with Python, 2nd Edition.* 2018, pág. 306, ISBN: 9788578110796. arXiv: arXiv:1011.1669v3.

[35] Y. Chen, X. P. Zhang, J. Yuan, B. Cai, X. L. Wang, X. L. Wu, Y. H. Zhang, X. Y. Zhang, T. Yin, X. H. Zhu, Y. J. Gu, S. W. Cui, Z. Q. Lu y X. Y. Li, "Association of body mass index and age with incident diabetes in Chinese adults: A population-based cohort study", *BMJ Open*, vol. 8, n.º 9, págs. 1-3, 2018, ISSN: 20446055. DOI: 10.1136/bmjopen-2018-021768.

[36] U. M. Learning, *Pima Indians Diabetes Database*, 2016. dirección: https://www.kaggle.com/uciml/pima-indians-diabetes-database.

[37] S. S. C. Xhan, *Expedientes Clínicos*, 2020.

[38] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann e I. H., "The WEKA Data Mining Software: An Update", *SIGKDD EXPLORATIONS*, vol. 11, n.º 1, págs. 10-18, 2009. dirección: `https://www.kdd.org/exploration%7B%5C_%7Dfiles/p2V11n1.pdf`.

[39] R. Joaquín Amat, *Precision, Recall, F1, Accuracy en clasificación*, 2020. dirección: `https://www.iartificial.net/precision-recall-f1-accuracy-en-clasificacion/` (visitado 01-10-2021).

[40] G. Garrett y H. Wickham, *R for Data Science*. O'Reilly Media, Inc., 2016, ISBN: 9781491910399. dirección: `https://www.oreilly.com/library/view/r-for-data/9781491910382/`.

[41] J. Mesa, *La hipertensión arterial en el paciente con diabetes tipo 2*, Barcelona, 2019. dirección: `https://www.revistaalad.com/`.

Anexo

En esta sección se darán a conocer trabajos realizados para la realización de ciertas fases dentro del trabajo, las cuales ayudaron a complementar los expedientes del conjunto de datos de México.

Anexo A

Docx: Sistema Web para la captura de pacientes y almacenamiento de diagnósticos

Docx es una extensión creada para el sistema SCoMedic, en la cual una persona registrada como doctor/médico tiene acceso desde cualquier dispositivo (teléfono, máquina o tableta). El objetivo de esta extensión es la de proveer al médico una manera versátil de poder tener información de sus pacientes con una conexión a internet dando el acceso a diagnósticos realizados con anterioridad dentro del sistema, resultados de una exploración física o antecedentes del paciente.

Al almacenar diagnósticos se toman en cuenta los estudios, en este caso se capturan seis estudios diferentes (glucosa, insulina, triglicéridos, colesterol total, colesterol *LDL* y colesterol *HDL*), pero la posibilidad de aumentar la cantidad de campos es ilimitada. Las imágenes mostradas (**Figura 6.1-6.2**) son extraídas de una base de datos de prueba dentro del sistema funcional pero se han censurado partes de la imagen para mantener la privacidad de los pacientes, dado a que la información es real.

El sistema también cuenta con la generación de fichas médicas personalizadas y de reportes de un diagnostico o consulta realizada mediante el sistema (**Figura 6.3**). Los

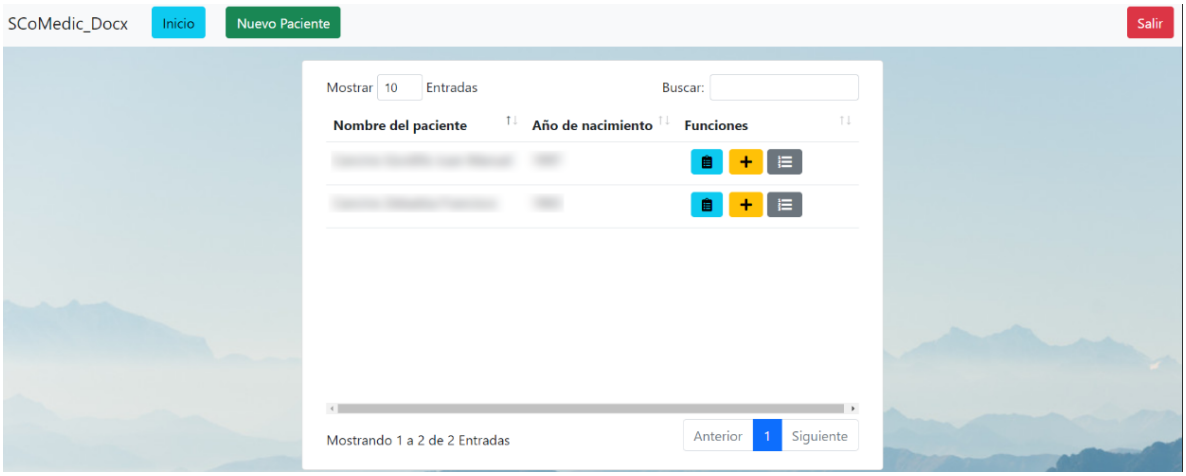


Figura 6.1: Pagina principal del sistema Docx.

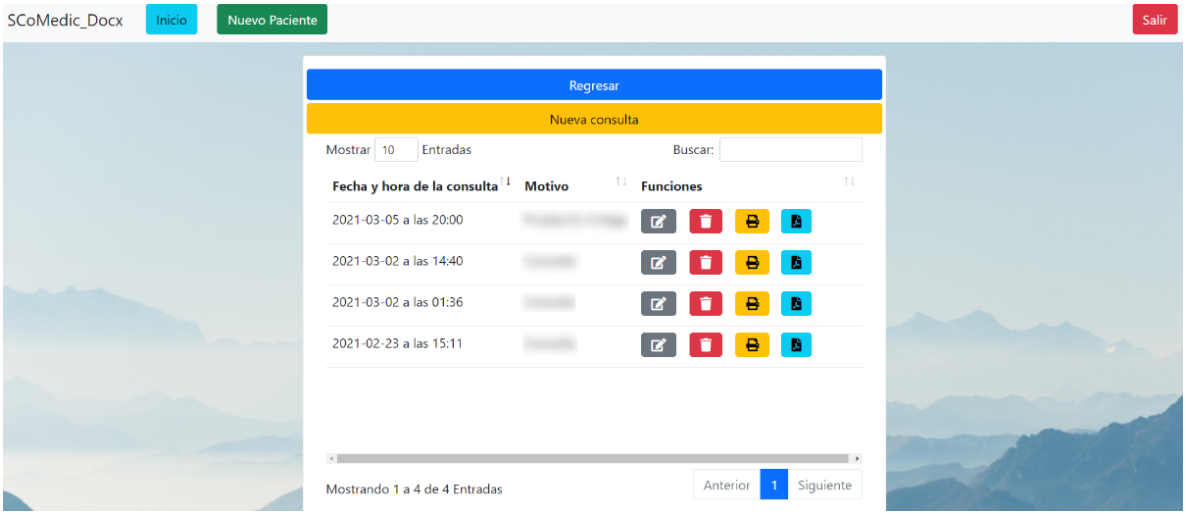
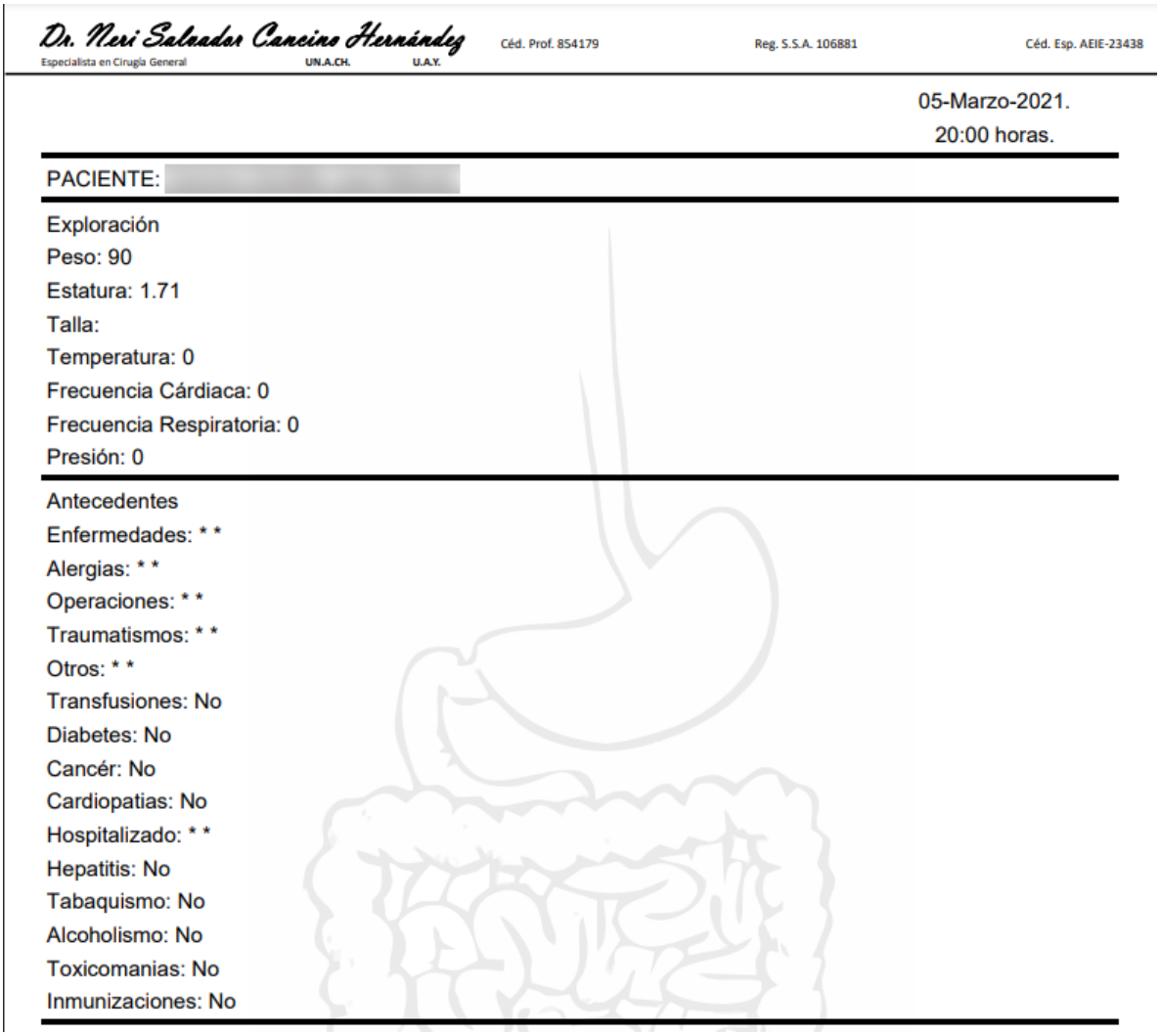


Figura 6.2: Consultas de un paciente.

campos con cero o asteriscos son campos que no se proporcionaron en dicha consulta.



Dr. Neri Salvador Cancino Hernández
Especialista en Cirugía General UN.A.CH. U.A.Y.
Céd. Prof. 854179 Reg. S.S.A. 106881 Céd. Esp. AEIE-23438

05-Marzo-2021.
20:00 horas.

PACIENTE:

Exploración
Peso: 90
Estatura: 1.71
Talla:
Temperatura: 0
Frecuencia Cárdiaca: 0
Frecuencia Respiratoria: 0
Presión: 0

Antecedentes
Enfermedades: * *
Alergias: * *
Operaciones: * *
Traumatismos: * *
Otros: * *
Transfusiones: No
Diabetes: No
Cancér: No
Cardiopatias: No
Hospitalizado: * *
Hepatitis: No
Tabaquismo: No
Alcoholismo: No
Toxicomanias: No
Inmunizaciones: No

Figura 6.3: Reporte generado automáticamente.

6.1. Anexo B

Recuperación de información dentro del historial clínico basado en índices invertidos

Para la recuperación de información de historiales clínicos se hace uso de la librearía *PyDoxc* para convertir un documento de Word a hipertexto etiquetado (*HTML*) para posteriormente hacer uso de la librería conocida como *BeautifulSoup* y navegar dentro del texto del archivo para finalmente generar un índice invertido con base a los textos recuperados y realizar la búsqueda de los parámetros dentro del índice.

La **Figura 6.4** muestra la interfaz del sistema, la cual cuenta de cuatro botones y un texto que nos indica la carpeta donde se está trabajando al uno de sus costados. El uso del programa es directo, seleccionando el directorio a trabajar como el primer paso, posteriormente se da clic a buscar y convertir archivos. El trabajo se realiza mediante recursividad para buscar dentro de la carpeta de trabajo cualquier documento con extinción Word (*docx*) y aplicar la librería *PyDoxc* y generar una carpeta con los archivos convertidos (**Listing 1**).

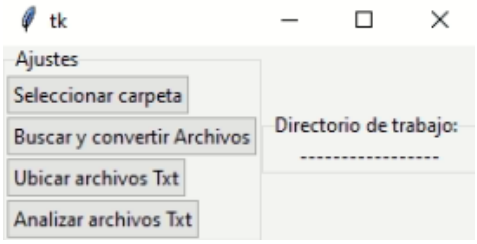


Figura 6.4: Interfaz del programa.

```
folder_to_track=directorio
   for path in os.listdir(folder_to_track):
       if (os.path.isfile(folder_to_track + "/" + path)):
           files = path.split(".")
           ext = files[-1].capitalize()
           if (ext == "Docx"):
               html = PyDocX.to_html(folder_to_track + "/" + path)
               escribirHTML(html,path+".txt")
       if (os.path.isdir(folder_to_track + "/" + path)):
           explorarPath(folder_to_track, path)
```

Listing 1: Example from external file

Una vez se seleccione la carpeta donde se encuentran los archivos *txt*, el programa mostrará salidas en la linea de comando donde se esta ejecutando. mostrando los valores encontrados dentro de los ficheros. El programa esta pensado en tomar los valores numéricos cercanos a la palabra que se esta buscando, tomando el ultimo valor valido como el valor que sera escrito en la tabla final. En el dado caso de no encontrar algún valor valido el usuario puede ingresar un número o simplemente continuar presionando la tecla *intro* (**Figura 6.5**).

```
--------------Buscando: Edad ---------------
edad 17 años
edad 20 años
edad
--------------Buscando: Sexo ---------------
No se encontro un valor valido para sexo Ingrese uno manualmente: M
--------------Buscando: Peso ---------------
No se encontro un valor valido para peso Ingrese uno manualmente:90
--------------Buscando: IMC ---------------
No se encontro un valor valido para imc Ingrese uno manualmente:
```

Figura 6.5: Texto mostrado en consola.

La **Figura 6.6** muestra la tabla resultante de la recopilación de información que es un concentrado de datos que posteriormente es almacenado en formato *csv* para ser combinado con el conjunto de datos de México. Para mantener la confidencialidad de los datos, el nombre del documento o del paciente nunca es mostrado durante el proceso de extracción de información o en el guardado de esta información.

Edad	Sexo	Estatura	Peso	IMC	PA Sistolica	PA Diastolica	Colesterol Total	Colesterol LDL	Colesterol HDL	Triglicéridos	Outcome
28	2	161	55	21.2	126	84					0
50	2	163	73	27.5	140	89	5.15			1.11	0
33	1	166	78.5	28.5	127	71	4.37			1.19	0
37	2	165	47	17.3	107	82	3.8	2.12	1.37	0.7	0
33	1	180	74	22.8	103	61	3.62			0.48	0

Figura 6.6: Concentrado de datos.

Al concatenar los datos hay que realizar un pequeño procesado en los datos, los cuales consisten en evitar los espacios en blanco dentro de los registros numéricos esto se hace con apoyo de la librería *pandas* usando un objeto llamado *dataframe* el cual nos da la posibilidad de usar la función *replace* (**Listing 2**) y cambiar los espacios en blancos por cualquier valor (en este caso se pone un 0).

```
data[['Edad','Peso','IMC','PA Sistolica',
'PA Diastolica',"Colesterol Total",
"Colesterol LDL","Colesterol HDL",
"Trigliceridos" ]] = data[['Edad','Peso',
'IMC','PA Sistolica','PA Diastolica',"Colesterol Total"
, "Colesterol LDL","Colesterol HDL","Trigliceridos"]].replace(" ","0")
```

Listing 2: Función replace de *pandas*