Universidad Autónoma del Estado de Hidalgo

Instituto de Ciencias Básicas e Ingeniería

Licenciatura en Ciencias computacionales

Proyecto Final

Minería de datos

Autores:
Katia Guadalupe Hernández García
Angel de Jesús Martínez Vega
Arantza Peña Hernández
Daniel Romero Resendiz

Índice general

Índice de figuras

Índice de cuadros

1

Introducción

La generación de datos en cantidades masivas en la actualidad es cada vez más frecuente en las instituciones que brindan algún servicio, estos datos pueden ser médicos, económicos, comerciales, educativos, científicos etc. Sin embargo, día a día estos grandes volúmenes de datos pertenecientes a empresas, instituciones, gobiernos y particulares crecen; esto dificulta enormemente el análisis constante de los datos por un experto para la identificación de un aspecto relevante de la misma.

Tradicionalmente el análisis de los datos para la obtención de conocimiento se llevaba a cabo por medio del análisis manual a través de un experto el cual se encargaba de llevar acabo un registro de los datos con lápiz y papel para ordenar y clasificar manualmente, implementado un amplio análisis estadístico, sin embargo, era un proceso lento y expuesto a generar errores debido a los grandes volúmenes de información escrita almacenada.

Debido al aumento de volumen en los datos, nos encontramos frente a un problema, la intoxicación, se dispone de tanta información, que a veces es imposible organizarla con efectividad y al surgimiento de la necesidad de emplear técnicas computacionales que fueran rápidas, baratas y objetivas en sus resultados para el análisis y obtención de conocimiento fue necesaria la implementación de Bases de Datos digitales que guardaran el histórico de los datos.

En la actualidad la revolución digital ha permitido que la información digitalizada sea más fácil de capturar, procesar, almacenar, distribuir, y transmitir por ello hoy en día es más fácil recoger y almacenar información en grandes bases de datos, sin embargo, para descubrir conocimiento de estos enormes volúmenes de datos es un reto en sí mismo.

Con la intención de ayudar a comprender y obtener conocimiento de la enorme cantidad de datos y que estos puedan contribuir a la mejora y crecimiento de las empresas, instituciones, gobiernos y particulares surgió la Minería de Datos según los autores Witten y Frank [11]. La Minería de datos es el proceso de extraer conocimiento útil y comprensible, previamente desconocido, desde grandes cantidades de datos almacenados en distintos formatos. El objetivo de la minería de datos es encontrar modelos a partir de los datos, utilizando procesos automáticos o semiautomáticos. Al llevar a cabo el minado de datos se descubrirán patrones los cuales apoyaran a la toma de decisiones más seguras, que beneficiaran al usuario final. En este siglo la demanda continuará creciendo, y el acceso a grandes volúmenes de datos multimedia traerá la mayor transformación para el global de la sociedad. Por tanto, el desarrollo de la tecnología de minería de datos avanzada continuará siendo una importante área de estudio, y en consecuencia se espera gastar muchos recursos en esta área de desarrollo en los próximos años.

Esencialmente, la minería de datos es un método innovador de aprovechar la información ya existente en las empresas, hospitales, instituciones gubernamentales y particulares a fin de, mejorar procesos, mejorar el rendimiento de la institución u optimizar el uso de recursos. Mediante la minería de datos, se pueden realizar consultas mucho más complejas los datos que utilizando métodos de consulta

convencionales. La información que la minería proporciona puede mejorar notablemente la calidad y fiabilidad de la toma de decisiones. Las herramientas de minería de datos facilitan y automatizan el proceso de descubrir esta clase de información en bases de datos de gran tamaño.

La minería de datos es una etapa del proceso KDD (Proceso de extracción de conocimiento en base de datos) la cual se refiere a la extracción no trivial de información implícita, previamente desconocida y potencialmente útil de los datos que se encuentran en las bases de datos dentro del proceso reiterativo de búsqueda o descubrimiento del conocimiento la minería de datos es el paso más importante sin embargo para llegar a él es necesario pasar por otra serie de etapas, estos pasos aplicados de una manera iterativa e interactiva aseguran que un conocimiento útil se extraiga de los datos.

Para convertir la enorme cantidad de datos a los que tiene acceso la industria bancaria, la comunidad médica, la industria comercial, los proveedores de servicios etc., en conocimiento útil es necesaria la implementación de la minería de datos , la minería ha permitido la resolución de problemas llevando a cabo el procesamiento automatizado de datos ya sean transaccionales, no operativos y metadatos, esto para obtener información (patrones, asociaciones o relaciones) , esta información se convierte en conocimiento valioso sobre patrones históricos y tendencias futuras lo cual nos permitirá obtener nuevos conocimientos y ventajas competitivas gracias a la utilización de diversas técnicas de minería de datos, ya sea de asociación, agrupamiento, clasificación, predicción, análisis de tendencias etc. Debido a que todos los días se procesan grandes cantidades de datos es necesario la implementación de la minería de datos para descubrir lo oculto aún.

El análisis de los datos mediante la Minería de Datos aporta numerosas ventajas en la obtención de conocimiento permite descubrir información que no esperábamos obtener, es capaz de analizar bases de datos con una enorme cantidad de datos, contribuye a la toma de decisiones tácticas y estratégicas para detectar la información clave, los algoritmos que se emplean en la etapa de minado de datos son confiables. Los modelos son probados y comprobados usando técnicas estadísticas antes de ser usado, para que las predicciones que se obtienen sean confiables y válidas la Minería de Datos trabaja mano a mano con los almacenes de datos, sobre todo en los casos de volúmenes de datos muy grandes o de interrelaciones entre los datos complejas.

La finalidad principal de la Minería de Datos es explorar, por medio de la utilización de distintas tecnologías y técnicas, enormes bases de datos de forma automática, con el propósito de encontrar patrones repetitivos, así como tendencias o reglas que expliquen el comportamiento de los datos que han sido recopilados en tiempo real.

Las técnicas de agrupamiento tratan de encontrar grupos entre un conjunto de individuos, el agrupamiento se basa en crear grupos de objetos que compartan características semejantes, todos estos objetos son descritos por un número determinado de variables o atributos. De un conjunto del que se desconoce una variable objetivo, con un algoritmo de agrupamiento se crean grupos disjuntos que ayuden a distinguir a objetos del conjunto total, para ello se hace uso de dos tipos de clasificación: supervisada y no supervisada.

La clasificación supervisada de datos es el proceso que se lleva a cabo para encontrar propiedades comunes entre un conjunto de datos y clasificarlos dentro de diferentes clases, de acuerdo con un modelo de clasificación. El objetivo de la clasificación es primero desarrollar una descripción o modelo para cada clase usando las características disponibles en los datos. Tales descripciones de las clases son entonces usadas para clasificar futuros datos de prueba en la base de datos o para desarrollar mejores descripciones (llamadas reglas de descripción) para cada clase en la base de datos. Las aplicaciones de la clasificación incluyen diagnóstico médico, predicción de rendimiento, mercadotecnia selectiva, por nombrar unas cuantas. Se parte de un conjunto de clases conocido a priori. Estas clases deben caracterizarse en función del conjunto de variables mediante la medición de estas en individuos cuya pertenencia a una de las clases no presente dudas (áreas de entrenamiento).

En la clasificación no supervisada no se establece ninguna clase a priori, aunque es necesario determinar el número de clases se quieren establecer, y se definen por un procedimiento estadístico. El aprendizaje no supervisado no presenta pares de entradas y salidas sino sólo la información de entrada que de alguna manera del algoritmo debe agrupar según sus parecidos razonables. La clasificación no supervisada se utiliza para la detección de patrones ocultos en bases de datos de gran tamaño. Su objetivo general es encontrar algún tipo de estructura en una colección de datos sin etiquetar o sin clasificar, ya que en la mayoría de los casos no se dispone de esta información.

En la medicina la Minería de Datos se utiliza para las grandes bases de datos con información de los pacientes todo se almacena en el historial, examen físico, patrones de terapias anteriores etc., con esta información y la implementación de técnicas de Minería de Datos, se es capaz de ofrecer diagnósticos más precisos. Además, permiten agilizar la gestión y los trámites administrativos, al tener toda la información bien segmentada y localizada.

Uno de los más grandes avances de la medicina en la actualidad con respecto al tema de la monitorización fetal fue sin duda la invención del registro cardiotocográfico (cardio, corazón; toco, parto). La cardiotocografía fetal mide la frecuencia cardíaca del feto durante el embarazo o el trabajo de parto y el parto, se lleva a cabo el registro de los cambios en la frecuencia cardíaca fetal y en su relación temporal con las contracciones uterinas. Estas mediciones pueden ayudar a los profesionales de la salud a verificar el estado general del feto y a identificar las primeras señales de sufrimiento fetal.

Durante el trabajo de parto y el parto, se observa de cerca la frecuencia cardíaca fetal junto con las contracciones uterinas de la madre. Esto ayuda al médico o a la enfermera a ver cómo responde el bebé y a saber si se necesita algún tratamiento, como el uso de medicamentos, para ayudar a acelerar el parto.

El objetivo de la cardiografía es a los fetos que pueden presentar una insuficiencia de oxígeno (hipoxia), por lo que se pueden utilizar evaluaciones adicionales del bienestar fetal o decidir extraer al feto mediante cesárea o parto vaginal instrumentado. El fundamento para realizar la MFE durante el trabajo de parto, es que los distintos tipos de trazados obtenidos son marcadores indirectos de la respuesta cardiaca y medular a los cambios de volumen, acidemia e hipoxemia, puesto que el cerebro modula la frecuencia cardiaca fetal.

En el presente proyecto se procesarán automáticamente 2126 cardiotocografías fetales (CTG) prenatal, registro electrónico continuo de la frecuencia cardíaca fetal que se obtuvo mediante un transductor de ultrasonido colocado sobre el abdomen materno. El conjunto de datos consta de mediciones de la frecuencia cardíaca fetal (FHR) y las características de contracción uterina (UC) esta información es obtenida del repositorio público [10]. Para las clases, la clasificación fue tanto a para un patrón morfológico como a un estado fetal. Para este estudio se utilizaron las 3 clases de estado fetal. El conjunto de datos se procesará pasando por cada una de las etapas del proceso KDD, preparación de los datos, minería de datos, evaluación e interpretación, difusión y uso de modelos. Haciendo uso del software Weka en la etapa de minería de datos para la obtención de conocimiento haciendo uso de los algoritmos de agrupamiento K-means , DBScan y finalmente Calinsky Harabasz con el objetivo de analizar cada una de las particiones resultantes del conjunto de datos a evaluar , de tal manera que se identifique cuál de los algoritmos ejecutados es más óptimo para la obtención de conocimiento el cual agrupara el conjunto de datos en las 3 clases de estado fetal normal, sospechoso , patológico logrando de esta manera ayudar a los profesionales de la salud a verificar el estado general del feto .

2

Marco Teórico

En el siguiente capitulo se explicará a fondo los métodos de agrupación que se realizaron para el desarrollo de este proyecto, además de la aplicación que permitió procesar la información

2.1. Knowledge Discovery in Databases (KDD)

Termino creado en 1989, se refiere al proceso de extracción de información desde una base de datos, procesarlos y por medio de ello obtener nueva información que servirá para el futuro. El proceso KDD está compuesto por varios pasos, para que de esta forma se complete de forma certera, siendo selección de datos, pre procesamiento y limpieza de los mismos, minería de datos e interpretación y evaluación de los mismos. Es importante recalcar la importancia de tener un proceso iterativo, siempre acompañado de un experto que sea capaz de comunicarnos cómo manejar la información en cualquier circunstancia que se nos presente [1].

El concepto KDD es relativamente nuevo por lo que su desarrollo permanece constante hoy en día, entrelazándose con áreas como base de datos, aprendizaje automático, reconocimiento de patrones, estadística, teoría de la información inteligencia artificial, razonamiento con incertidumbre, visualización de datos y computación de altas presentaciones. Una vista general sobre los sistemas KDD se pueden encontrar en las siguiente referencia. [2].

En general el proceso KDD consiste en procesar el conocimiento de grandes cantidades de datos, además de la capacidad de almacenar estos datos, tener acceso de forma veloz, aplicar algoritmos a estos e interpretarlos. Hoy en día conocemos el valor que lo datos tienen y como a través de estos tenemos capacidad predictiva, sumamente valiosa ya sea para el sector comercial, médico, etc. [3].

2.2. Minería de datos

Recordando, la minería de datos es parte del proceso KDD, una vez obteniendo la vista minable de los datos logrando que los datos tengan valor, traduciéndolo a conocimiento. Posteriormente es necesario saber que se desea obtener de los datos que queremos procesar, de esta manera es posible escoger el modelo o relación a partir de los datos más convenientes. Para utilizar el modelo más conveniente es importante tomar en cuenta puntos como: de donde se obtuvieron los datos, la cantidad de datos a procesar, la funcionalidad que se le dará después del proceso y tipos de aplicación[3].

2.3. Algoritmos no supervisados

En la minería de datos hay dos algoritmos principales, los supervisados y los no supervisados, ambas categorías son capaces de encontrar patrones ocultos en grandes conjuntos de datos.
Los modelos no supervisados son conocidos como descriptivos o no dirigidas, los algoritmos de estos modelos basan su proceso de entrenamiento en una función de datos no etiquetados o clases previamente definidas.
Los algoritmos no supervisados no se enfocan en atributos predeterminados, tampoco realiza predicciones a valores específicos, sino que encuentra estructuras y relaciones ocultas entre los datos.

En estos algoritmos la distancia es sumamente importante, ya que a través de esta podemos agrupar objetos más similares, o en este caso cercanos. En el momento en que el valor de la distancia es muy pequeño, es cuando tenemos la certeza de agrupación. Existen diversas fórmulas capaces de medir distancia de acuerdo a la necesidad del algoritmo, sin embargo, la más conocida es la distancia euclidiana.

2.3.1. Algoritmos por agrupación

Las funciones de agrupación son de las más comunes para la exploración de los datos, por medio de algoritmos de minería de datos de agrupamiento se pueden encontrar agrupaciones naturales.
El agrupamiento o análisis de agrupamiento es el proceso de poner objetos en grupos cuyos miembros son similares, por lo que un paso importante en los métodos de agrupación es definir la similitud o la distancia. Un buen método de agrupación genera grupos de alta calidad, para garantizar que la similitud entre grupos sea baja y la similitud interna del grupo sea alta[4].

2.3.2. Algoritmo K-means

El algoritmo de agrupamiento K-means, es uno de los algoritmos más simples y eficientes propuestos en la minería de datos. Cuando se presenta un conjunto de datos sin etiquetas, una de las tareas más sencillas que podemos realizar es encontrar grupos de datos en nuestro conjunto de datos que sean similares entre sí, lo que es conocido como grupos.
El algoritmo K-means comienza eligiendo puntos representativos como centros iniciales, después cada punto se asigna al más cercano. El centro está basado en una medida de proximidad particular elegida. Una vez que se formen las agrupaciones, se actualizan los centros de cada grupo. Posteriormente, el algoritmo se repite iterativamente hasta que los centros no cambien o se cumpla cualquier otro criterio alternativo de una agrupación relajada [5].
La distancia euclidiana entre los grupos $x = (x_1, x_2, ..., x_i)$ y $y = (y_1, y_2, ..., y_i)$ la formula seria la siguiente[6].

$$d(x_i, y_i) = \sqrt{\sum_{n}^{i=1}(x_i - y_i)^2} \tag{2.1}$$

En la siguiente ecuación (2.2) x_i simboliza la alineación central del grupo ci y x simboliza el objeto, que es objetivo. E es el total del error cuadrático de todos los puntos. La función de criterio, que mide la distancia y se utiliza para encontrar la distancia de cada elemento del conjunto y su distancia al centro es la distancia euclidiana.

$$E = \sum_{i=1}^{k} \sum_{x \in c_i} |x - x_i|^2 \tag{2.2}$$

Ejemplo

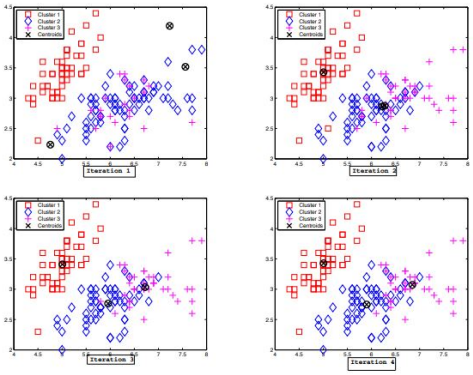


Figura 2.1: Ilustración de 4 iteraciones de K-means sobre el conjunto de datos Fisher Iris

En la Figura 2.1 se proporciona una ilustración de las diferentes etapas de la ejecución del algoritmo

de 3 medias en el conjunto de datos "Fisher Iris". La primera iteración inicializa tres puntos aleatorios como centros. En iteraciones posteriores, los centros cambian de posición hasta el agrupamiento. Se puede usar una amplia gama de medidas de proximidad dentro de las K-medias[5].

2.3.3. Algoritmo Jerárquico

Los algoritmos de agrupamiento jerárquico se desarrollaron para construir un mecanismo más detallista y flexible para agrupar los objetos de datos. Los algoritmos jerárquicos tienen por objetivo agrupar para formar un grupo nuevo o bien separar alguno ya existente para dar origen a otros, de tal forma que, si sucesivamente se va realizando este proceso de división, se minimice alguna distancia o bien se maximice alguna medida de similitud. El algoritmo Funciona mediante la agrupación de datos en un árbol de agrupaciones con estadísticas de agrupamiento jerárquico al tratar cada punto de datos como un grupo individual. El punto final se refiere a un conjunto diferente de agrupaciones, donde cada grupo es diferente del otro grupo y los objetos dentro de cada grupo son iguales entre sí[7].
Los métodos jerárquicos se subdividen en aglomerativos y disociativos. Cada una de estas categorías presenta una gran diversidad de variantes. Los métodos aglomerativos toman agrupaciones únicas, estas contienen un objeto de datos por grupo, comienzan en el nivel inferior y continúan fusionando dos grupos a la vez para construir una jerarquía ascendente de los grupos.Los métodos divisivos comienzan con todos los objetos de datos en un gran grupo y lo dividen continuamente en dos grupos.

Métodos jerárquicos aglomerativos

Los pasos involucrados en un algoritmo de agrupamiento jerárquico aglomerativo son los siguientes: Primero, usando una medida de proximidad particular, se construye una matriz de disimilitud y todos los datos los puntos se representan visualmente en la parte inferior de un dendrograma, los conjuntos agrupados más cercanos son fusionados en cada nivel y luego la matriz de disimilitud se actualiza correspondientemente, hasta llegar a la raíz [5].

Proceso del método aglomerativo

- Determinar la similitud entre los individuos y todos los demás grupos. (Encontrar matriz de proximidad)
- Considerar cada punto de datos como un grupo individual
- Combinar grupos similares.
- Recalcular matriz de proximidad para cada grupo.
- Repetir los pasos 3 y 4 hasta que obtenga un solo grupo

Ejemplo En el siguiente ejemplo siguiendo la figura 2.2 Supóngase que se tienen seis puntos de datos:

- Se considera una letra como un solo grupo y se calcula la distancia de un grupo de los demás grupos
- Los grupos comparables se fusionan para formar un único grupo
  - Para mezclar los grupos, por ejemplo sí el representante del grupo es $C_a = c_{a1}, c_{a2}, ..., c_{an}$ y del grupo b es $C_b = c_{b1}, c_{b2}, ..., c_{bn}$, y a tiene j objetos y b tiene k objetos el nuevo representante se calcula:

$$C = \left\{ \frac{j * c_{a1} + k * c_{b1}}{j+k}, \frac{j * c_{a2} + k * c_{b2}}{j+k}, \frac{j * c_{an} + k * c_{bn}}{j+k} \right\} \tag{2.3}$$

- Se recalcula la proximidad y se fusionan los dos grupos más cercanos

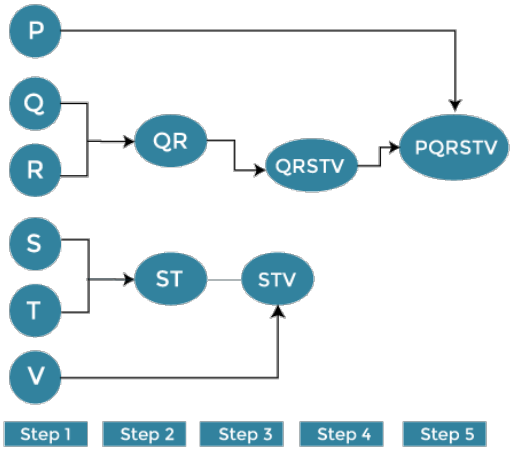


Figura 2.2: Agrupación jerárquica divisiva

- Los dos grupos restantes se fusionan para formar un solo grupo [(PQRSTV)].

[7].

Métodos jerárquicos divisivos

El agrupamiento jerárquico divisivo es un enfoque de arriba hacia abajo en el que el procedimiento comienza en la raíz con todos los puntos de datos y lo divide recursivamente para construir el dendrograma, Puede ser considerado como un enfoque global ya que contiene la información completa antes de dividir los datos, todos los puntos de datos se consideran un grupo individual y, en cada iteración, los puntos de datos que no son similares se separan del grupo. Los puntos de datos separados se tratan como un grupo individual. Finalmente, nos quedamos con N grupos [5].

2.3.4. Algoritmos basados en densidad

Los algoritmos de agrupamiento, los más conocidos, suponen que los datos se generan a partir de una distribución de portabilidad de un tipo determinado, debido a esta suposición, estos algoritmos producen grupos esféricos y no se pueden tratar bien con conjuntos de datos en los que los grupos reales tienen formas no esféricas. Los tamaños de las bases de datos de la vida real son cada vez más grandes, la agrupación de grandes bases de datos requiere la capacidad de detectar y eliminar ruido al igual, que los valores atípicos. El paradigma del agrupamiento basado en la densidad ha sido propuesto para abordar todos estos requisitos. La agrupación basada en la densidad se puede considerar como un método no perimétrico, ya que no hace suposiciones sobre el número de grupos o su distribución [5].

DBSCAN

Utiliza una distancia específica para separar los grupos densos del ruido más escaso. El algoritmo DBSCAN es el más rápido de los métodos de agrupamiento, pero solo es apropiado si hay una distancia de búsqueda muy clara para usar, y eso funciona bien para todos los agrupamientos potenciales. Esto requiere que todos los grupos significativos tengan densidades similares. Este método también le permite usar los parámetros, campo de tiempo e intervalo de tiempo de búsqueda para encontrar grupos de puntos en el espacio y el tiempo[5].

En el vecindario E es fundamental para DBSCAN para aproximar la densidad local, por lo que el algoritmo tiene dos parámetros:

- E: Es radio de nuestros vecindarios, alrededor de un punto de datos.

- minPts: El número mínimo de puntos de datos que queremos en un vecindario para definir un clúster.

Utilizando estos dos parámetros, DBSCAN clasifica los puntos de datos en tres categorías:

- Puntos centrales: Un punto con tantos o más vecinos como el número mínimo de puntos.
- Puntos de borde: Un punto que tiene menos vecinos que el número mínimo de puntos, pero es vecino de un punto central.
- Valor atípico: Un punto de datos o es un valor atípico si no es ni un punto central ni un punto de borde.

DBSCAN estima la densidad contando el número de puntos en una vecindad de radio fijo y considera que dos puntos están conectados si se encuentran dentro de la vecindad del otro. Un punto se llama punto central si la vecindad de radio contiene una cantidad mayor o igual de MinPts, es decir, la densidad en la vecindad tiene que exceder algún umbral. Un punto q es alcanzable por la densidad desde un punto central de p si q está dentro de la vecindad de p. Un grupo es entonces un conjunto de puntos conectados por densidad que es máxima con respecto a la densidad-alcanzable. El ruido se define como el conjunto de puntos de la base de datos que no pertenecen a ninguno de sus grupos. La tarea de la densidad basada en agrupamiento es encontrar todos los grupos con respecto a los parámetros épsilon y MinPts en una base de datos dada[5].

2.4. Índice de validez

El índice de validez es un problema al que uno se enfrenta en el agrupamiento, es decidir la partición óptima de los datos en grupos. En este contexto, la visualización del conjunto de datos es una verificación crucial de los resultados del agrupamiento. La validez de agrupamientos consiste en un conjunto de técnicas para encontrar un conjunto de agrupamientos que se ajuste mejor a las particiones naturales, sin ninguna información de clase a priori. El resultado del proceso de agrupación se valida mediante un índice de validez de agrupación. La validación externa y la validación interna son las dos categorías más importantes para la validación de agrupamiento. A diferencia de las técnicas de validación externas, las técnicas de validación interna miden el agrupamiento únicamente basadas en información de los datos, evalúan que tan buena es la estructura del agrupamiento sin necesidad de información ajena al propio algoritmo y su resultado.

Es de gran importancia evaluar el resultado de los algoritmos de agrupamiento, sin embargo, es difícil definir cuando el resultado de un agrupamiento es aceptable. Por esta razón existen técnicas e índices para la validación de un agrupamiento realizado [8].

2.4.1. Tipos de validación

La validación externa y la validación interna son las dos categorías más importantes para la validación de agrupación. La principal diferencia es si se usa o no información externa para la validación, es decir, información que no es producto de la técnica de agrupación utilizada.

Podemos observar en la figura anterior que cuando el modelo tiene una complejidad baja, los errores de entrenamiento y prueba pueden coincidir, en cambio cuando el error de predicción es alto, mientras aumenta la complejidad del modelo, disminuye el error de prueba. Esta disminución de error de prueba llega hasta cierto punto, en el que aumenta la complejidad del modelo hasta que el modelo alcanza un error mínimo de prueba (predicción), de esta forma alcanzará la complejidad óptima del modelo. Para concluir mejor, obtenemos que los algoritmos supervisados son algoritmos de optimización en la cual, con ayuda de una función, minimizan cierto funcional que representa el error de predicción del modelo.

Figura 2.3: Gráfico de líneas de los valores de CH frente al número de grupos para el conjunto de datos

2.4.2. Calinski-Harabasz Index

El índice de Calinski-Harabasz se puede utilizar para evaluar el modelo cuando no se conocen las etiquetas verdaderas del terreno donde la validación analizará que tan bien se ha ejecutado el método de agrupamiento, utilizando cantidades y características propias al conjunto de datos. El índice CH, también conocido como criterio de varianza es una medida que muestra que tanto hay de similitud entre un objeto a su propio grupo (cohesión) en comparación de otros grupos (separación). La cohesión considera las distancias desde los puntos de datos en un grupo hasta su centro y la separación se basa en la distancia de los centros del grupo desde el centro global[9].

Cálculo del Índice de Calinski-Harabasz: El índice CH para K número de conglomerados en un conjunto de datos:

$$CH = \frac{N - K \sum_{C_K \in C}^{|C|} |C_k| d_e(\bar{c_k}, \bar{x})}{K - 1 \sum_{C_k \in C}^{|C|} \sum_{X_i \in C_k}^{|C_k|} d_e(x_i, \bar{c_k})} \tag{2.4}$$

Donde, n_k y c_k son los números de puntos y el centro del grupo k, c es el centro global, n es el número total de puntos de datos.
En el gráfico de la figura 2.3 se puede observar que se necesita elegir aquella solución que dé como resultado un pico en el gráfico de líneas de los índices CH. Por otro lado, si la línea es suave entonces no hay razón para preferir una solución sobre otras. Un valor más alto del índice CH significa que los grupos son densos y están bien separados [9].

2.5. Algoritmos supervisados

También llamados predictivos, son aquellos que predicen un dato (o un conjunto), comenzamos con datos desconocidos y a partir de ellos obtenemos datos conocidos. Estos algoritmos tienen dos fases. La primera fase, llamada fase de entrenamiento o supervisión trabaja con un conjunto de datos que servirán para entrenamiento, posteriormente los parámetros internos se ajustarán de forma que se minimice el error de predicción de la variable dependiente. En la siguiente fase, llamada fase de entrenamiento se aplica la fase de prueba en la cual se realizará la estimación del error en el modelo de prueba, (no en el de supervisión). El error encontrado será una aproximación más cercana. El objetivo de este algoritmo es encontrar modelos que minimicen el error de predicción. En la siguiente figura encontraremos un gráfico en donde se muestra el comportamiento de los errores de entrenamiento en diferentes complejidades.

2.5.1. Clasificación Bayesina

Clasificadores estadísticos, capaces de predecir las probabilidades del número de miembros de clase, así como la probabilidad de que una muestra pertenezca a una clase particular. Se basa en el teorema de Bayes, el cual a demostrado una alta exactitud y velocidad en entornos con una gran cantidad de datos. Incluso ha sido comparado con algoritmos como árboles de decisión y clasificadores de redes de

neuronas, teniendo como resultado un buen rendimiento.

El clasificador Naive Bayesiano quiere saber a través del aprendizaje cuál es la mejor hipótesis de acuerdo a los datos que procesa. Se utiliza una variable para marcar la probabilidad a priori de los datos (los datos más probables a obtener) en este caso se denotará como P(D). Además, se utilizará la probabilidad de los datos dada una hipótesis, denotada como P(h|D). Por último, el valor que se desea encontrar es la probabilidad de h, procesando los datos introducidos con anterioridad. Para representar este valor se utilizara P(h|D), todo lo encontraremos en la siguiente ecuación[12].

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} \tag{2.5}$$

En caso que h tenga k posibles valores, se debe de identificar el valor mayor y devolverlo como resultado de la clasificación. De esta manera se obtendrá el valor de clase que tenga la máxima probabilidad a posteriori dados los atributos, esto es llamado como Hipótesis máxima a posteriori (MAP).

$$C_{MAP} = p(c) \quad \prod_{i=1}^{n} \; p(\quad X_i|c) \tag{2.6}$$

La ecuación anterior debe de aplicarse k veces, una por cada clase. Como se mencionó anteriormente el valor mayor será la clase elegida. También, la letra n representa el número de atributos del problema a tratar aportando directamente a la probabilidad.

2.5.2. K-NN

Considerado como uno de los mejores representantes de este tipo de aprendizaje. Como su nombre lo indica, esté algoritmo es capaz de encontrar los objetos más cercanos al objeto que se desea clasificar y de esta forma asignarle una clase. Al igual que muchos otros algoritmos este utilizara funciones de distancia, como euclidiana o Manhattan. En el ejemplo en la figura 2.4 podemos encontrar como a se clasifica como -, y b como + [12]..

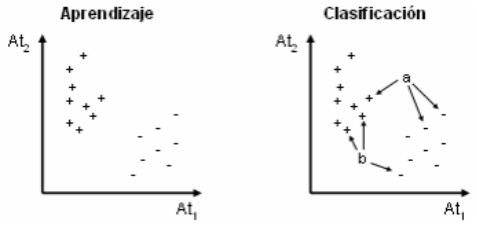


Figura 2.4: Ejemplo de Aprendizaje y Clasificación con KNN

En este algoritmo el objeto a evaluar se asignará a los vecinos más cercanos en una región.

3

Weka

Weka es un software desarrollado por ka universidad de Waikato (Nueva Zelanda), tiene una alta colección de algoritmos de Maquinas de conocimiento que pueden ser aplicados sobre datos mediante varias interfaces que ofrece. Esta aplicación también tiene la capacidad de realizar transformaciones sobre datos, realizar clasificación, regresión, clustering, asociación y visualización. Una gran ventaja que proporciona Weka es su diseño, ya que es una herramienta extensible y añadir nuevas funcionalidades no es complicado en la ecuación [13].

3.1. Ficheros .arff

Weka trabaja con este formato, acrónimo de Attribute-Relation File Format. Se encuentra compuesto en una estructura de tres partes

1. Cabecera: Nombre de la relación, expresada con datos tipo String, por lo que si se requieren espacios en blanco es necesario agregar el entrecomillado

@relation <nombre-de-la-relacion>

2. Declaraciones de atributos: Atributos que tendrá el archivo junto a su tipo de dato. El nombre del atributo será tipo String. Serán con 5 tipos de datos, ya sea Numeric en donde nos encontraremos con números reales, Integer, en donde toparemos con números enteros

3. Sección de datos: Datos que componen la relación separando entre comas los atributos y con saltos de línea las relaciones. @data 4,3.2

A continuación un ejemplo de este lenguaje:

```
@attribute Género {Hombre,Mujer}
@attribute Rama {Humanidades,Sociales,Ciencias,Salud,Ingeniería_arquitectura}
@attribute edad integer
@attribute n_hermanos integer
@attribute Rendimiento real

@data
Hombre,Humanidades,19,2,6.7
Hombre,Salud,25,?,9.1
Mujer,Salud,20,1,7.3
Hombre,Sociales,21,3,?
Hombre,Humanidades,20,0,5.0
```

Figura 3.1: Ejemplo lenguaje

3.2. Interfaz principal

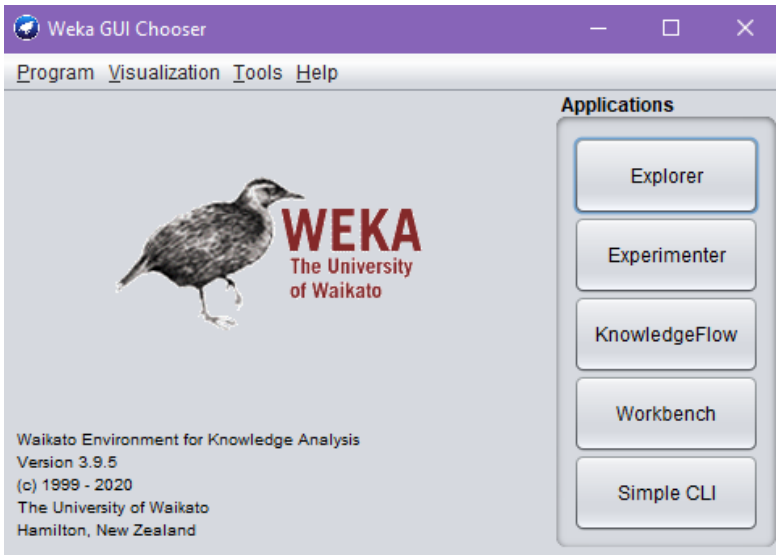


Figura 3.2: Interfaz Weka

- Explorer: Esta es una interfaz visual WEKA para trabajos gráficos simplemente. En este modo, puede procesar, clasificar, vincular y visualizar sus datos. Sencillo e intuitivo con un solo archivo de datos.
- Experimenter: Una manera conveniente de aplicar uno o más métodos de clasificación Método automático. Esta ventana facilita la realización de grandes experimentos. escala.
- KnowledgeFlow: Esta es una GUI y se utiliza para el desarrollo de proyectos. Flujo de información.
- Simple CLI: Esto se denomina interfaz de línea de comandos y se utiliza para la invocación Directamente al paquete Java incluido en WEKA. Esto es más conveniente para nosotros tan pronto como se abre la ventana del explorador.

3.3. Parámetros de agrupación

1. K-means

- canopyMinimumCanopyDensit: Si usa una agrupacion canopy para inicializar con un mínimo de densidad T2
- DisplayStdDevs: Mostrar la desviación típica además de la media en los valores asignados en cada variable a cada agrupacion.
- DistanceFunction: Método para calcular las distancias entre los sujetos y los centroides, normalmente Distancia euclidiana.
- InitializationMethod:: Punto de partida para iniciar el algoritmo, normalmente de manera aleatoria, tomando una semilla.
- fastDistanceCalc: Para reducir el tiempo de cómputo, se calculan las distancias a partir de puntos de corte en lugar de las sumas de cuadrados de las distancias.
- dontReplaceMissingValues: Remplaza los valores faltantes mediante la media/moda.
- canopyPeriodicPruningRate: Si usas agrupación canopy, se usa para la inicialización cuando deseas bajar la densidad durante el entrenamiento

- canopyMaxNumCanopiesToHoldInMemory: si se utiliza la agrupación de canopy para inicialización y/o aceleración este es el número máximo de canopies candidatos para retener en la memoria principal durante el entrenamiento del agrupador.
- canopy T2: La distancia que se usa cuando agrupas canopy. Valores <0 indican que puede usarse la heurística basada en atributos con desviación estándar.
- doNotCheckCapabilities: Las capacidades del agrupador no se chequea antes de que empiece dicha agrupación.
- preserveInstancesOrder: Preservar el orden de las instancias.
- initializationMethod: El método de inicialización a usar. Aleatorio, k-medias++,Canopy o el más lejano primero
- maxIterations: Máximo de iteraciones permitidas en las que el modelo debe converger
- numClusters: Número de agrupaciones que serán establecido.
- numExecutionSlots: Número de ranuras de expansión empleadas para el cómputo (depende de cada ordenador, si no se sabe, mantener 1)
- speed: El numero aleatorio con el que se inicializará
- fastDistanceCalc: utiliza valores límite para acelerar el cálculo de la distancia, pero suprime también el cálculo y la salida de la suma dentro del grupo de cuadrados errores/suma de distancias.
- reduceNumberOfDistanceCalcsViaCanopies: use la agrupación de canopy para reducir el número de cálculos de distancia realizados por k-means.

A continuación se mostrará una imagen con los parámetros por defecto que tiene Weka.

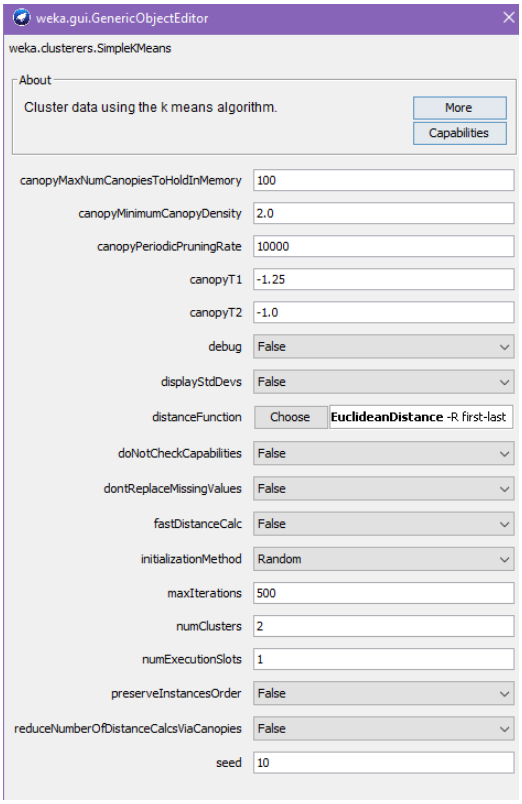


Figura 3.3: Parámetros K-means

2. DBScan

- debug:Si se establece en verdadero, el clasificador puede generar información adicional en la consola
- DistanceFunction: Método para calcular las distancias entre los sujetos y los centroides, normalmente Distancia euclidiana.
- doNotCheckCapabilities: Las capacidades del agrupador no se chequea antes de que empiece dicha agrupación.
- épsilon: Establece el valor de épsilon en el que se ejecutara el algoritmo.
- minPoints: Establece el mínimo de puntos para realizar agrupaciones.

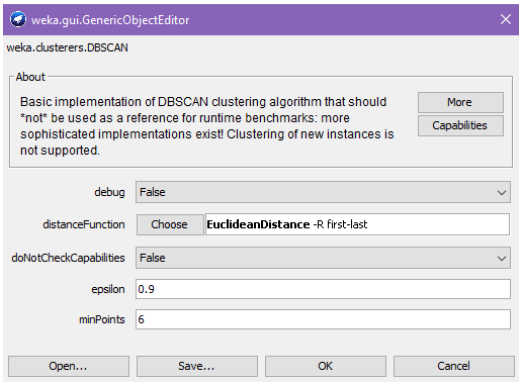


Figura 3.4: Parámetros DBScan

3.4. Parámetros de clasificación

En este proyecto se utilizaron los siguientes clasificadores:

2. NaiveBayes

- batchSize: Si se realiza la agrupación de los sujetos por lotes, tamaño de cada lote.
- debug:Si se establece en verdadero, el clasificador puede generar información adicional en la consola
- doNotCheckCapabilities: Las capacidades del agrupador no se chequea antes de que empiece dicha agrupación.
- useKernelEstimator: Se usa para estimar probabilidades en las tablas de una red de Bayes segun Kernel.
- numDecimalPlaces: Numero de decimales que tomará en cuenta[13].

2. KNN

- KNN: Numero de vecinos más cercanos a considerar.
- batchSize: Si se realiza la agrupación de los sujetos por lotes, tamaño de cada lote
- debug:Si se establece en verdadero, el clasificador puede generar información adicional en la consola
- CrossValidation: Resultados fruto de una validación cruzada (Falso/Verdadero)

- doNotCheckCapabilities: Las capacidades del agrupador no se chequea antes de que empiece dicha agrupación.
- numDecimalPlaces: Número de decimales presentados en los datos mostrados en el output.
- numDecimalPlaces: Numero de decimales que tomará en cuenta[13].

4

Resultados

En este capítulo se analizaron los resultados de los diversos algoritmos relacionados a la minería de datos, fueron aplicados al conjunto de datos, Cardiotocography Data Set, esté fue ejecutado en el software WEKA. En el transcurso de este capítulo veremos más a fondo el conjunto de datos y los resultados arrojados.

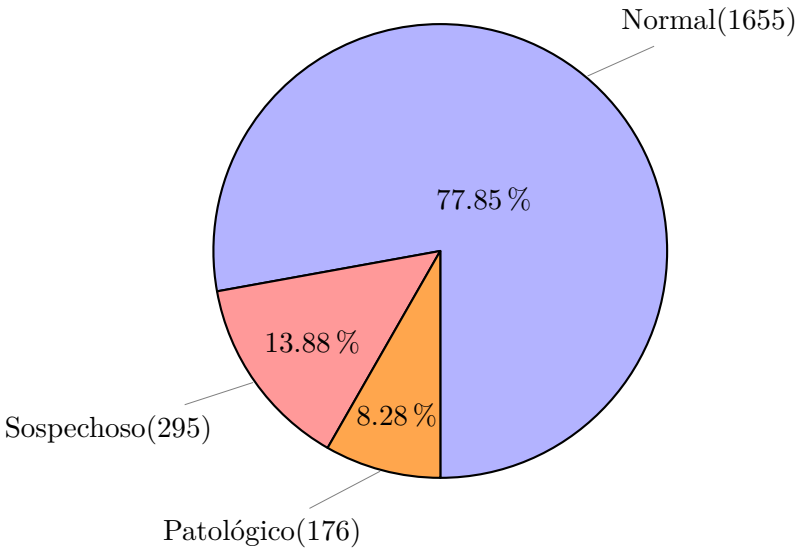
Atributos	
Atributo	Descripción
LBE	Valor de referencia (El experto médico)
LB	Valor de línea de base (latidos por minuto)
AC	Aceleraciones por segundo
FM	Movimiento fetal por segundo
UC	Contracciones uterinas por segundo
ASTV	Porcentaje de tiempo con variabilidad anormal a corto plazo
mSTV	Valor medio de la variabilidad a corto plazo
ALTV	Porcentaje de tiempo con variabilidad anormal a largo plazo
mLTV	Valor medio de la variabilidad a largo plazo
DL	desaceleraciones de leves
DS	desaceleraciones severas
DP	desaceleraciones prolongadas
DR	desaceleraciones repetitivas
Width	ancho de histograma
Min	Baja frecuente del histograma
Max	alta frecuente del histograma
Nmax	número de picos de histograma
Nzeros	número de ceros de histograma
Mode	modo de histograma
Mean	media de histograma
Median	mediana de histograma
Variance	varianza de histograma
Tendency	Tendencia del histograma: -1 = Assimétrico izquierdo; 0 = simétrico; 1 = Assimétrico derecho

Cuadro 4.1: Descripción de atributos

4.1. Cardiotocography Data Set

El siguiente conjunto de datos se procesan y clasifican cardiotocografías, las cardiotocografías son exámenes para evaluar el bienestar fetal, el conjunto de datos se compone de mediciones de frecuencia cardiaca fetal y las características de contracción uterina clasificadas por obstetras expertos.
El número de objetos presentes son 2126, los cuales se procesaron automáticamente cardiotocografías fetales y se midieron las características de diagnóstico respectivos.

El número de atributos son 23, Como podemos ver en la tabla 4.1 los atributos describen diferentes características como, el número de latidos, las aceleraciones por segundo, movimientos fetales, contracciones uterinas, desaceleraciones, tiempo, propiedades de histogramas, etc. Para las clases, la clasificación fue tanto a para un patrón morfológico como a un estado fetal. Para este estudio se utilizaron las 3 clases de estado fetal. NSP código de clase de estado fetal (N=normal; S=sospechoso; P=patológico)



4.2. K-means

Al comienzo se agrupo los datos con el método K-Means, utilizando valores de 2, 3 y 5, siendo los más óptimos para este grupo de datos. Para esto se agregan los datos a una tabla, que posteriormente se pasan a un formato csv. De está manera el documento puede ser abierto en un programa de texto. Posteriormente en el editor de texto agregamos el nombre de relación que se está realizando siendo "@RELATION cardio", numeramos los atributos siendo "@ATTRIBUTE x NUMERIC", agregamos la clase siendo "@ATTRIBUTE class 1,2,3 ɛ finalmente agregamos "@DATAïndicando que a partir de ahí se encontrarán todos los datos que procesaremos. Finalmente agregaremos el formato .arff para poder abrirlo en Weka

En la siguiente tabla 4.2 se encontrarán los resultados que obtuvimos una vez que procesamos los datos en Weka

Agrupaciones								
–	CLUSTER	Grupos	Ruido	g0	g1	g2	g3	g4
Partición 1	2	2	0	1206	920	-	-	-
Partición 2	3	3	0	832	582	712	-	-
Partición 3	5	5	0	511	442	468	312	393

Cuadro 4.2: Resultados del algoritmo de K-means

4.3. DBSCAN

Para este algoritmo se realizó en un gran número de agrupaciones con diferentes mínimos de puntos y valores de épsilon. Conforme se agrupaba se utilizaron muchísimos valores para encontrar los más acertados y de está manera encontrar las mejores particiones. En el desarrollo de este algoritmo se incluyen los datos clasificados como "no supervisados". En la siguiente tabla 4.3 se encuentran las diferentes particiones realizadas, es necesario recalcar que las 3 mejores particiones encontradas se subrayaron.

Agrupaciones												
Epsilon	Min p	Grupos	Ruido	g0	g1	g2	g3	g4	g5	g6	g7	g8
0.019	8	0	2126	-	-	-	-	-	-	-	-	-
0.046	3	6	2107	4	3	3	3	3	3	-	-	-
0.052	3	6	2107	4	3	3	3	3	3	-	-	-
0.057	4	1	2122	4	-	-	-	-	-	-	-	-
0.057	3	7	2122	4	3	3	3	3	3	3	-	-
0.059	4	1	2122	4	-	-	-	-	-	-	-	-
0.06	3	8	2099	4	3	5	3	3	3	3	3	-
0.07	3	9	2093	4	3	8	3	3	3	3	3	-
0.075	4	2	2116	4	6	-	-	-	-	-	-	-
0.075	3	8	2093	4	11	3	3	3	3	3	3	-
0.087	3	15	2070	4	4	4	3	11	3	3	3	3
0.088	4	3	2107	4	4	11	-	-	-	-	-	-
0.09	15	0	2126	-	-	-	-	-	-	-	-	-
0.097	4	5	2099	4	4	4	4	11	-	-	-	-
0.099	4	6	2095	4	4	4	4	11	4	-	-	-
0.109	4	11	2074	4	5	4	4	4	4	11	4	4
0.1099	7	0	2126	-	-	-	-	-	-	-	-	-
0.152	6	5	2087	6	13	6	6	8	-	-	-	-
0.18	6	12	2020	8	8	7	13	7	6	6	8	12
0.18	7	5	2078	7	13	8	8	12	-	-	-	-
0.18	8	4	2085	13	8	8	12	-	-	-	-	-
0.18	9	2	2104	13	9	-	-	-	-	-	-	-
0.1888	6	18	1966	8	9	6	8	6	13	8	6	-
0.1888	8	4	2069	8	13	27	9	-	-	-	-	-
0.1888	10	2	2099	13	14	-	-	-	-	-	-	-
0.1889	7	7	2048	7	8	13	7	27	7	9	-	-
0.189	7	7	2048	7	8	13	7	27	7	9	-	-
0.189	8	4	2069	8	13	27	9	-	-	-	-	-
0.19	8	4	2069	8	13	27	9	-	-	-	-	-
0.195	7	11	2015	8	8	13	7	7	29	7	8	7
0.1959	7	12	2006	8	8	13	7	7	7	11	28	7
0.197	8	6	2052	8	8	13	28	8	9	-	-	-
0.1978	7	12	1997	8	8	13	7	7	7	7	35	11
0.1979	8	6	2052	8	8	13	28	8	9	-	-	-
0.1988	7	12	1997	8	8	13	7	7	7	7	35	11
0.1988	9	6	2050	8	8	13	28	8	11	-	-	-
0.1989	8	6	2050	8	8	13	28	8	11	-	-	-
0.199	7	12	1997	8	8	13	7	7	7	7	35	11
0.199	10	3	2087	13	13	13	-	-	-	-	-	-
0.2	11	3	2092	11	11	12	-	-	-	-	-	-
0.222	11	3	2067	11	19	29	-	-	-	-	-	-
0.228	10	9	1973	17	11	10	-	-	-	-	-	-
0.235	11	7	1960	16	11	13	19	80	12	15	-	-
0.2375	11	7	1960	16	11	13	19	80	12	15	-	-
0.2342	10	8	1913	26	11	91	10	10	29	13	19	14
0.2348	10	8	1913	26	11	91	10	10	29	13	19	14
0.2428	10	6	1843	229	5	12	10	13	14	-	-	-
0.2482	12	5	1887	27	159	12	28	13	-	-	-	-
0.2527	11	4	1824	266	12	13	11	-	-	-	-	-
0.255	12	3	1860	241	12	13	11	12	15	-	-	-

Cuadro 4.3: Resultados del algoritmo DBSCAN Pt.1

Agrupaciones												
Epsilon	Min p	Grupos	Ruido	g0	g1	g2	g3	g4	g5	g6	g7	g8
0.2558	13	7	1889	24	134	28	13	11	12	15	-	-
0.256	11	4	1809	281	12	13	11	-	-	-	-	-
0.2561	9	15	1632	315	14	16	15	8	10	16	13	11
0.2562	10	12	1709	292	5	10	13	15	10	15	13	10
0.262	11	4	1827	263	12	13	11	-	-	-	-	-
0.2728	12	10	1638	351	15	33	13	12	16	12	12	12
0.282	10	11	1440	486	43	10	38	10	14	13	13	35
0.289	12	8	1504	463	55	12	13	15	13	34	17	-
0.298	12	6	1438	413	64	64	39	21	13	38	-	-
0.311	12	7	1276	717	12	14	12	47	36	12	-	-
0.3119	12	7	1228	739	12	44	17	14	14	11	46	-
0.3131	8	15	1021	831	9	8	13	111	29	14	12	13
0.331599	11	7	1195	781	14	49	14	14	11	48	-	-
0.318	9	15	1033	847	13	58	10	14	10	11	10	11
0.32	10	11	1094	830	13	55	14	10	11	12	51	10
0.325	12	6	1155	827	14	55	14	12	49	-	-	-
0.319	8	14	945	910	10	8	123	28	14	13	13	9
0.3232	12	6	1167	816	13	55	14	12	49	-	-	-
0.3359	12	6	1145	838	14	55	14	12	48	-	-	-
0.333	11	7	1042	1007	14	13	11	11	11	17	-	-
0.3356	12	7	1061	936	12	65	14	14	12	12	-	-
0.345	9	11	800	1188	12	10	21	11	15	12	10	10
0.358	10	6	753	1296	10	14	31	10	12	-	-	-
0.34	9	13	814	1100	9	10	39	21	16	12	15	12
0.3599	10	4	729	1354	10	14	10	12	-	-	-	-
0.3649	11	4	769	1266	65	15	11	-	-	-	-	-
0.36511	10	7	658	1394	20	10	10	14	10	12	-	-
0.3683	10	8	636	1404	20	8	10	10	14	10	14	-
0.376	11	5	676	1389	23	12	15	11	-	-	-	-
0.3782	9	7	519	1527	22	9	14	13	13	9	-	-
0.38	10	7	558	1486	22	9	14	13	14	10	-	-
0.398	6	6	275	1819	6	6	6	8	6	-	-	-
0.432	7	3	209	1902	7	8	-	-	-	-	-	-
0.4011	12	4	459	1621	18	13	15	-	-	-	-	-
0.40345	12	4	445	1638	18	13	12	-	-	-	-	-
0.41	7	2	270	1848	8	-	-	-	-	-	-	-
0.41	6	5	242	1858	6	6	8	6	-	-	-	-
0.42	6	5	207	1893	6	6	8	6	-	-	-	-
0.432	7	3	209	1902	7	8	-	-	-	-	-	-
0.433	6	5	177	1923	6	6	8	6	-	-	-	-
0.48	4	9	55	2022	7	8	6	6	6	5	6	5
0.48	5	9	67	2014	5	6	6	6	6	5	6	5
0.48	6	2	146	1972	8	-	-	-	-	-	-	-
0.48	8	2	146	1972	8	-	-	-	-	-	-	-
0.48	9	2	151	1966	9	-	-	-	-	-	-	-
0.48	8	2	146	1972	8	-	-	-	-	-	-	-
0.488	6	5	94	2008	6	6	6	6	-	-	-	-
0.488	4	9	52	2025	7	8	6	6	6	5	6	5
0.4888	6	5	91	2011	6	6	6	6	-	-	-	-
0.4888	4	9	51	2026	7	8	6	6	6	5	6	5

Cuadro 4.4: Resultados del algoritmo DBSCAN Pt.2

Agrupaciones												
Epsilon	Min p	Grupos	Ruido	g0	g1	g2	g3	g4	g5	g6	g7	g8
0.49	4	9	48	2029	7	8	6	6	6	5	6	5
0.49	5	9	57	2023	5	7	6	6	6	5	6	5
0.49	6	5	87	2015	6	6	6	6	-	-	-	-
0.498	7	2	110	2009	7	-	-	-	-	-	-	-
0.5005	6	5	71	2030	6	7	6	6	-	-	-	-
0.5065	7	2	102	2017	7	-	-	-	-	-	-	-
0.518	4	6	43	2055	6	6	5	6	5	-	-	-
0.51924	5	7	44	2048	6	6	6	5	6	5	-	-
0.534	5	6	40	2057	6	6	6	5	6	-	-	-
0.55	4	6	31	2068	6	6	5	6	4	-	-	-
0.55	5	5	37	2066	6	6	5	6	-	-	-	-
0.55	6	5	43	2059	6	6	6	6	-	-	-	-
0.56	5	5	35	2068	6	6	5	6	-	-	-	-
0.56	6	4	40	2068	6	6	6	-	-	-	-	-
0.57	6	4	38	2070	6	6	6	-	-	-	-	-
0.58	4	8	20	2071	4	6	6	5	6	4	4	-
0.58	5	5	32	2071	6	6	5	6	-	-	-	-
0.58	6	4	37	2071	6	6	6	-	-	-	-	-
0.588	6	4	37	2071	6	6	6	-	-	-	-	-
0.59	6	4	36	2072	6	6	6	-	-	-	-	-
0.6	3	8	17	2075	3	4	6	6	5	6	4	-
0.6009	12	1	62	2064	-	-	-	-	-	-	-	-
0.601	10	1	57	2069	-	-	-	-	-	-	-	-
0.6015	8	1	54	2072	-	-	-	-	-	-	-	-
0.6038	4	7	20	2075	4	6	6	5	6	4	-	-
0.60468	5	5	28	2075	6	6	5	6	-	-	-	-
0.611	5	5	27	2076	6	6	5	6	-	-	-	-
0.613	2	12	8	2076	3	2	4	2	6	6	5	6
0.613	3	8	16	2076	3	4	6	6	5	6	4	-
0.615	3	7	15	2081	3	4	6	6	5	6	-	-
0.6246	8	1	50	2076		-	-	-		-	-	-
0.6246	5	6	19	2079	6	6	5	6	5	-	-	-
0.633	3	6	14	2086	3	6	6	5	6	-	-	-
0.634	5	6	18	2080	6	6	5	6	5	-	-	-
0.635	2	10	5	2087	3	2	2	6	6	5	6	2
0.6399	3	6	12	2088	3	6	6	5	6	-	-	-
0.64	5	6	16	2082	6	6	5	6	5	-	-	-
0.64	2	10	4	2088	3	2	2	6	6	5	6	2
0.6401	2	10	4	2088	3	2	2	6	6	5	6	2
0.6405	4	5	15	2088	6	6	5	6	-	-	-	-
0.64468	10	1	45	2081	-	-	-	-	-	-	-	-
0.651	2	7	4	2099	2	6	5	6	2	2		
0.653	5	5	15	2089	6	5	6	5	-	-	-	-
0.663	4	4	10	2099	6	5	6	-	-	-	-	-
0.679	3	3	9	2106	5	6	-	-	-	-	-	-
0.6809	12	1	38	2088	-	-	-	-	-	-	-	-
0.696	9	1	31	2095	-	-	-	-	-	-	-	-
0.697	6	4	13	2094	7	6	6	-	-	-	-	-
0.699	5	3	7	2108	5	6	-	-	-	-	-	-

Cuadro 4.5: Resultados del algoritmo DBSCAN Pt.3

Agrupaciones												
Epsilon	Min p	Grupos	Ruido	g0	g1	g2	g3	g4	g5	g6	g7	g8
0.7	11	4	51	283	1646	79	67	-	-	-	-	-
0.715	10	4	47	284	1674	79	69	-	-	-	-	-
0.7185	12	4	47	284	1674	79	69	-	-	-	-	-
0.722	11	4	46	284	1647	80	69	-	-	-	-	-
0.728	10	3	46	284	1647	149		-	-	-	-	-
0.732	12	4	51	283	1646	79	67	-	-	-	-	-
0.738	11	4	284	1647	81	69		-	-	-	-	-
0.742	12	4	283	1646	79	67		-	-	-	-	-
0.748	7	4	36	285	1653	152		-	-	-	-	-
0.752	15	4	44	285	1647	83	67	-	-	-	-	-
0.7555	9	3	40	285	1649	152		-	-	-	-	-
0.7562	8	3	40	285	1649	152		-	-	-	-	-
0.7568	16	4	45	285	1647	83	67	-	-	-	-	-
0.7628	13	4	40	285	1649	83	69	-	-	-	-	-
0.7655	15	4	43	285	1647	84	67	-	-	-	-	-
0.768	11	3	40	285	1649	152		-	-	-	-	-
0.772	15	3	42	285	1647	152		-	-	-	-	-
0.7755	17	4	43	285	1647	81	70	-	-	-	-	-
0.778	16	4	43	285	1647	81	70	-	-	-	-	-
0.786	17	4	43	285	1647	81	70	-	-	-	-	-
0.792	15	3	41	285	1647	153		-	-	-	-	-
0.798	12	3	39	285	1649	153			-	-	-	-

Cuadro 4.6: Resultados del algoritmo DBSCAN Pt.4

4.4. Calinsky Harabasz

Finalmente, con el algoritmo Calinsky Harabasz se puede obtener, de acuerdo a las seis particiones obtenidas de los distintos algoritmos, cuales son más efectivos. A continuación, observaremos en las tablas 4.7 y 4.8 en donde se muestran los resultados, recordando que en este algoritmo el valor más alto demuestra la mejor partición.

K-means			
–	Grupos	Ruido	Calinsky Harabasz
Partición 1	2	0	195.53
Partición 2	3	0	1256.76
Partición 3	5	0	737

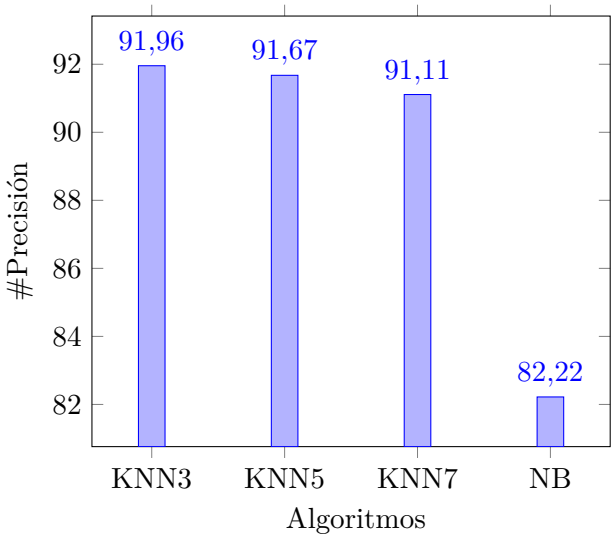
Cuadro 4.7: Resultados del algoritmo K-means con el índice de validez Calinsky Harabasz

DBScan					
–	Epsilon	Min. Pts	Grupos	Ruido	Calinsky Harabasz
Partición 1	0.38	10	7	558	28.66
Partición 2	0.48	4	9	55	10.42
Partición 3	0.289	12	8	1504	45.06

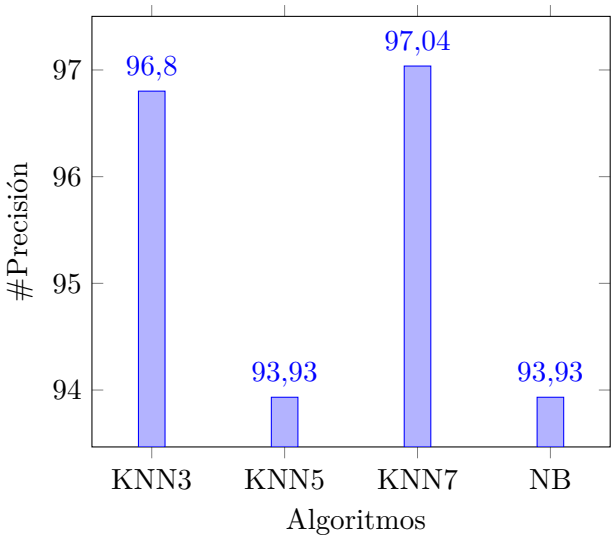
Cuadro 4.8: Resultados del algoritmo DBScan con el índice de validez Calinsky Harabasz

4.5. Algoritmos de clasificación

En el gráfico se puede visualizar la precisión arrojado por medio del conjunto de datos con clasificación original, al cual se le aplicaron algoritmos de clasificación, los algoritmos que se utilizaron fueron Naive Bayes y k-nearest neighbors (KNN) donde varia su valor en k de 3, 5 y 7.



Para el siguiente gráfico , se utilizó un conjunto de datos sintético, del cual de las 6 particiones obtenidas por medio de los algoritmos de agrupación y al que se proceso por medio del índice de validez Calinsky-Harabasz, se obtuvo un agrupamiento óptimo, a esté resultado se le aplicaron los algoritmos de clasificación, Naive Bayes y k-nearest neighbors(KNN) con las variaciones del valor en k de 3, 5 y 7.



Bibliografía

[1] Timarán-Pereira, S. R., Hernández-Arteaga, I., Caicedo-Zambrano, S. J., Hidalgo-Troya, A. y Alvarado Pérez, J. C. (2016). El proceso de descubrimiento de conocimiento en bases de datos. En Descubrimiento de patrones de desempeño académico con árboles de decisión en las competencias genéricas de la formación profesional (pp. 63-86). Bogotá: Ediciones Universidad Cooperativa de Colombia. doi: http://dx.doi.org/10.16925/9789587600490

[2] J. Hernández-Orallo, M. J. Ramírez-Quintana and C. Ferri. Introducción a la Minería de Datos. Prentice Hall / Addison-Wesley, 2004.

[3] Riquelme, J. C., Ruiz, R. y Gilbert, K. (2006). . Introducción. En Minería de Datos: Conceptos y Tendencias: Vol. 10. 29 (pp. 11–18).) Departamento de Lenguajes y Sistemas Informáticos Universidad de Sevilla. https://www.redalyc.org/pdf/925/92502902.pdf

[4] Taft, M., R. K., Mark, H., Denis, M.,George, T. (June de 2005). Oracle Data Mining Concepts 10g Release 2 (10.2). Obtenido de `https://docs.oracle.com/cd/B19306_01/datamine.102/b14339.pdf`

[5] Aggarwal, C. C., and Reddy, C. K. (2013). Data Clustering : Algorithms and Applications. Obtenido de `https://haralick.org/ML/data_clustering.pdf`

[6] Muhammet Esat OZDAG, E. C. (s.f.). Discovery of Course Success Using Unsupervised. Obtenido de MOJET: https://files.eric.ed.gov/fulltext/EJ1283323.pdf

[7] s.a. (s.f.). javatpoint. Obtenido de Clustering jerárquico en minería de datos: `https://www.javatpoint.com/hierarchical-clustering-in-data-mining#:~:text=Hierarchical%20clustering%20refers%20to%20an,points%20as%20an%20individual%20cluster.`

[8] Vazirgiannis M. (2009) Clustering Validity. In: LIU L., ÖZSU M.T. (eds) Encyclopedia of Database Systems. Springer, Boston, MA. `https://doi.org/10.1007/978-0-387-39940-9_616`

[9] s.a. (25 de abril de 2022). geeksforgeeks. Obtenido de Índice de Calinski-Harabasz: índices de validez de conglomerados | conjunto 3: `https://www.geeksforgeeks.org/calinski-harabasz-index-cluster-validity-indices-set-3/`

[10] UCI Machine Learning Repository: Physicochemical Properties of Protein Tertiary Structure Data Set. (s. f.). `https://archive.ics.uci.edu/ml/datasets/Physicochemical+Properties+of+Protein+Tertiary+Structure`

[11] H.Witten y E.Frank, (2011). Data Mining Practical Machine Learning Tools and Techniques (3.a ed.). Morgan Kaufmann Publishers. `https://www.wi.hs-wismar.de/~cleve/vorl/projects/dm/ss13/HierarClustern/Literatur/WittenFrank-DM-3rd.pdf`

[12] Molina, J. M. y Garcia, J. (s. f.). Data Mining. En Tecnicas de Mineria de Datos basadas en Aprendizaje Automatico (Vol. 3) `https://santiagozapatakdd.files.wordpress.com/2011/03/curso-kdd-full-cap-3.pdf`.

[13] Garcia Morate, D. (s. f.). Manual de Weka. Doctorado Formación en la Sociedad del Conocimiento | Universidad de Salamanca. `https://knowledgesociety.usal.es/sites/default/files/MANUAL%20WEKA.pdf`