Universidad Autónoma de Querétaro

Facultad de Informática

Maestría en Sistemas de Información: Gestión y Tecnología

Clasificación de datos en un espacio multidimensional: un enfoque sistémico de soporte a la toma de decisiones

TESIS

Que como parte de los requisitos para obtener el grado de

Maestro en Sistemas de Información: Gestión y Tecnología

Presenta:

Elieth Velázquez Chávez

Dirigido por:

Dr. Arturo González Gutiérrez

Universidad Aut6noma de Querétaro

Facultad de Informática

Maestría

Clasificación de datos en un espacio multidimensional: un enfoque sistémico de soporte a la toma de decisiones

Presenta:
Elieth Velazquez Chavez

Dirigido por:
Dr. Arturo Gonzalez Gutierrez

Dr. Arturo Gonzalez Gutierrez Presidente

Firma

Dr. Jaime Rangel Mondrag6n Secretario

Firma

M.C. Fidel Gonzalez Gutierrez Vocal

Firma

Dr, Alberto Pastrana Palma Suplente

Firma

M.C. Guillermo Diaz Delgado Su

plente

Firma

M.C. Ruth Angélica Rico Hernández
Director de la Facultad de Informática

Dr. Luis Gerardo Hernández Sandoval
Director de Investigación y Posgrado

Centro Universitario Queretaro, Qro.
Marzo 2010
Mexico

RESUMEN

El estudio y modelado de los Sistemas de Soporte a las Decisiones (SSD) han atraído la atención de la comunidad académica desde hace varias décadas. Una de las definiciones dada por Sprague y Carlson, establece que un SSD es un sistema interactivo basado en computadoras, que asiste a los tomadores de decisiones, utilizando datos y modelos, para resolver problemas semi-estructurados y no estructurados dentro de una organización. Con la finalidad de desarrollar SSD más efectivos, los desarrolladores de modelos de negocios y los investigadores se han interesado en las técnicas de minería de datos, las cuales permiten, mediante el reconocimiento de patrones, descubrir información valiosa escondida en los datos. Uno de los retos que hacen interesante al problema y justifican el estudio de técnicas de reconocimiento de patrones es el manejo de grandes cantidades de datos. Para ello, se utilizan técnicas de agrupamiento (clustering) de datos. El trabajo consiste en identificar estructuras o subclases de los objetos almacenados en las bases de datos que tengan un significado relevante en su campo de aplicación. La ventaja principal al usar las técnicas de agrupamiento de datos es la posibilidad de descubrir estructuras o categorías bien definidas y relevantes directamente en los datos sin tener que contar con un conocimiento preciso del contexto. En el presente trabajo se presentan dos casos de estudio. El primero utiliza una base de datos de estudiantes aspirantes a la Facultad de Informática, y tiene el propósito de encontrar las características que distinguen a los aspirantes aceptados de los no aceptados. Se cuenta con una base de datos de 5,500 registros correspondientes al periodo de 2004-1 hasta 2008-2. Las herramientas para éste análisis son Mathematica y SQL Server. El segundo caso de estudio consiste en el problema de localización de grupos de interés de maestros y materias. A partir de información representada mediante una matriz binaria de 20 x 20. En ambos casos de estudio se emplean técnicas de clustering y de optimización.

(**Palabras clave**: Clustering, clasificación de datos, soporte a la toma de decisiones)

SUMMARY

The study and modeling of the Decision Support System (DSS) has drawn the attention of academic community since many decades ago. One of the most significant definitions given so far by Sprague and Carlson, established that a DSS is an interactive system based on computers that helps decision makers to solve semi-structured and non-structured problems by using models and data within an organization.

Looking for develop more efective SSD the business model people and researches are interested in dataminning techniques, because its lets pattern reconognition and find important hidden infomation.

One of the first challenges that makes the problem interesting and justifies the study of pattern recognition techniques of databases in DSS is the huge amount of data. Thus, clustering techniques such as data clustering are used. The problem relies on identifying structures or subclasses from the stored objects in a database with a relevant significance in its application field. The main advantage of clustering data techniques is the possibility of finding very well defined and relevant structures or categories from the data without taking into account context expertise.

In this project two study cases are presented. The first one consists of a candidate student database from an academic institution, and the goal is to find characteristics that make a clear difference between candidates who are accepted and the candidates who are not. We have a database with of around 5,500 registers in the period from 2004-1 to 2008-2. The tools for the analysis are Mathematica and SQL server. The second study case is about finding groups in a matrix of 20x20 with binary entries. We use optimization and clustering techniques in both of the study cases.

(**Key words**: decision support systems, clustering, data classification)

A mis padres por su enorme apoyo,
a Leonardo y Rodrigo por que inspiran cada momento en mi vida.

AGRADECIMIENTOS

Agradezco a los sinodales por su tiempo y dedicación para guiar este trabajo en especial al Dr. Arturo González Gutiérrez. A los compañeros maestros que no solo me apoyaron con sus comentarios alentadores si no que también con su amistad. A mis padres que sin su ayuda esto no habría sido posible.

Í n d i c e

Índice de Tablas

Tabla.....Página

Índice de Figuras

FIGURA ..... PÁGINA

FIGURA ..... PÁGINA

1 Introducción

LA SED PARA EL CONOCIMIENTO ES UNA CARACTERÍSTICA HUMANA NATURAL. ARISTÓTELES

A lo largo de la historia se ha manifestado interés por la recolección y análisis de datos. El calendario maya, los dibujos de la cueva de Árdales, la escritura cuneiforme de las tablillas babilónicas, y la piedra Rosetta son ejemplos de la necesidad humana de transcendencia a fin de dar sentido a la vida en el mundo a través del registro de datos mediante el uso de símbolos (Barry 2001). Se puede decir también que los primeros indicios de búsqueda de información en los datos han sucedido al estudiar los símbolos de las antiguas civilizaciones con el fin de descifrarlos. Así ha sido posible darnos una idea de la manera en que vivían, cazaban, convivían, y al final poder aprender de las culturas pasadas para comprender un poco nuestro presente.

Actualmente este mismo proceso de registrar y analizar datos a fin de encontrar información relevante que sea la base para que las personas u organizaciones tomen mejores decisiones se ha dado en llamar *minería de datos* (DM, *Data Mining*). La DM tiene sus raíces entonces en una de las más antiguas actividades humanas: el deseo de dejar plasmada la experiencia en alguna forma (numérica o simbólica), para describirla y preservarla. Tan pronto como describimos y preservamos la experiencia a través de los datos, comenzamos el proceso inevitable de interpretarlos, utilizando alguna técnica de DM.

En la época actual, después de más de 60 años de cómputo comienza a ser importante el análisis de la información que se ha venido generando. Ahora en México por ejemplo, podemos recuperar información y obtener actas de nacimiento de manera ágil, procedimientos que antes tomaban varios días, semanas y en ocasiones hasta meses. La iniciativa de e-gobierno que es parte del Plan Nacional de Desarrollo del período 2001-2003 (http://pnd.fox.presidencia.gob.mx/) constituye un momento importante que ha motivado la digitalización de todos los libros de registro de nacimiento que se

tenían almacenados en los archivos del Registro Civil. Este suceso lleva entonces a un paso importante donde se comienza a analizar la información que se tiene, y así empezar a entender fenómenos sociales, naturales, planear mejor nuestros proyectos de ciudades, analizar riesgos, entre otras muchas oportunidades que puede dar el estudiar a fondo la información almacenada.

La DM se puede definir como el proceso que facilita la extracción no trivial de información que reside de manera implícita en los datos. Dicha información es previamente desconocida y podrá resultar útil para algún otro proceso. En otras palabras, la DM prepara, sondea y explora los datos para obtener la información oculta en ellos. Es importante señalar que DM no es una técnica, sino que ese nombre engloba a todo un conjunto de técnicas. El proceso de DM está fuertemente ligado a la supervisión de procesos industriales, ya que resulta muy útil cuando los datos almacenados en las bases de datos son procesados.

Los fundamentos de la DM se encuentran en las áreas de las ciencias computacionales y el análisis estadístico. Mediante los modelos basados en algoritmos de DM es posible generar la solución a problemas de predicción, clasificación y segmentación.

El algoritmo de DM es el mecanismo que crea un modelo de minería de datos. Para crear un modelo, un algoritmo analiza primero un conjunto de datos y luego busca patrones y tendencias específicos. El algoritmo utiliza los resultados de este análisis para definir los parámetros del modelo de DM. A continuación, estos parámetros se aplican en todo el conjunto de datos para extraer patrones procesables y estadísticas detalladas. El DM crea un algoritmo que puede tomar diversas formas, incluyendo:

1. Un conjunto de reglas que describen cómo se agrupan los productos en una transacción.
2. Un árbol de decisión que predice si un cliente determinado comprará un producto.
3. Un modelo matemático que predice las ventas.

4. Un conjunto de clusters que describen cómo se relacionan los casos de un conjunto de datos.

Algunos de los tipos de algoritmos de DM se pueden clasificar como lo muestra la figura 1:1.

- Algoritmos de clasificación, que predicen una o más variables discretas, basándose en otros atributos del conjunto de datos. Un ejemplo de algoritmo de clasificación es el de árboles de decisión.
- Algoritmos de regresión, que predicen una o más variables continuas, como las pérdidas o los beneficios, basándose en otros atributos del conjunto de datos. Un ejemplo de algoritmo de regresión es el de serie temporal.
- Algoritmos de segmentación, que dividen los datos en grupos, o clústeres, de elementos que tienen propiedades similares. Un ejemplo de algoritmo de segmentación es el de clustering.
- Algoritmos de asociación, que buscan correlaciones entre diferentes atributos de un conjunto de datos. La aplicación más común de esta clase de algoritmo es la creación de reglas de asociación, que pueden utilizarse en un análisis de compras en sistemas del tipo comercio electrónico.

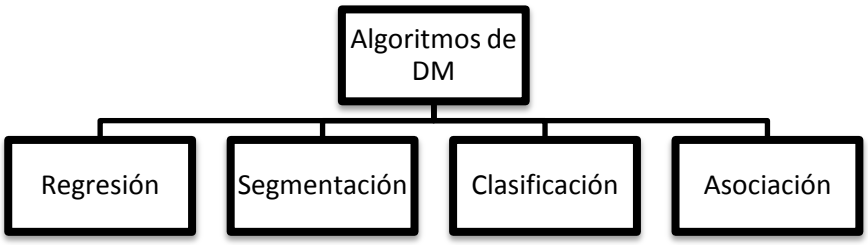


FIGURA 1:1 ALGORITMOS DE MINERÍA DE DATOS

La actividad de explotación de datos creció a la par con el incremento masivo de su almacenamiento. Al principio los datos eran pasivos, útiles en las funciones de facturación, pero no se explotaban más allá de la idea de únicamente contar con información histórica.

Los datos requieren ser dinámicos para poder establecer que les sea inherente un elemento de predicción, lo que es útil en la toma de decisiones de un negocio. Este concepto llevó al nacimiento de los sistemas tipo EIS (*Executive Information Systems*). Y éstos, en conjunto con los sistemas para toma de decisiones dieron lugar al concepto de Inteligencia de Negocio (Power 2007).

El concepto de Inteligencia de Negocios (*Business Intelligence, BI*) involucra conceptos de organización de datos en varias dimensiones de análisis a efecto de poder visualizar los datos.

A manera de ejemplo, consideremos una base de datos con información de las ventas que se están teniendo en la empresa. Es posible organizar los datos a fin de poder visualizarlos desde el punto de vista de ventas por región, pero quizás sería también interesante analizarla desde la perspectiva de los productos vendidos. Entre más dimensiones se incluyan en el análisis es mejor la visualización de cómo se comporta el negocio, y así fundamentar la toma de decisiones.

Es de llamar la atención que alrededor del término DM diferentes autores presentan también los términos: *Data Warehousing, BI, DSS*. La bibliografía varía, desde la que está dirigida para los empresarios, tratando de convencerlos acerca de la importancia de implementar dichas técnicas en sus empresas, hasta la que está dirigida a los arquitectos de software.

Actualmente en el mercado existen herramientas comerciales y también herramientas de software libre y arquitectura abierta que incluyen funciones que efectúan procesos de minería de datos. Entre las herramientas comerciales se encuentran las propias de manejadores de bases de datos como Oracle con su OBI Suite (http://www.bi-dw.info/oracle.htm) e IBM Cognos (http://www-01.ibm.com/software/data/cognos/) Business Intelligence and

Financial Performance Management. En cuanto a software libre se cuenta por ejemplo, con Pentaho BI Suite (http://www.pentaho.com/) entre otros.

A menudo, encontramos en el mercado de software de sistemas que las empresas venden servicios de DW, BI y SSD como la solución a cualquier problema empresarial.

En el presente trabajo, se utilizó la herramienta MS SQL Server 2008, debido a su facilidad de uso y a la cobertura de aplicaciones que la empresa Microsoft posee.

1.1 Antecedentes

El estudio y modelado de los SSD han atraído la atención de la comunidad académica desde hace varias décadas. Una de las definiciones dadas por Sprague y Carlson, (Yu 2004), establece que un SSD es un sistema interactivo basado en computadoras, que asiste a los tomadores de decisiones, utilizando datos y modelos, para resolver problemas semi-estructurados (parte del problema tiene una respuesta dada por un procedimiento aceptado) y no estructurados (no cuentan con un procedimiento definido) dentro de una organización. Sin embargo, el rápido avance de los sistemas basados en Internet y el reciente crecimiento de los sistemas de Comercio Electrónico (*Electronic Commerce, CE*), han sido propulsores de nuevas estrategias y procesos para llevar a cabo los negocios, y por ende han propiciado una redefinición de los SSD. Así, muchos modelos innovadores de negocios han emergido a partir del CE, y algunos han buscado apoyarse en los SSD, explotando la información que se recupera de la Web.

Un tema de interés y pieza clave en las tecnologías Web es la minería Web, que consiste básicamente en minería de datos aplicada a datos provenientes de la Web. Particularmente, la minería del comportamiento de la Web ha estado en el centro de atención de la comunidad científica y tecnológica. Las preguntas relevantes que se han establecido refieren al análisis del comportamiento de los usuarios, al descubrimiento de patrones de comportamiento del usuario, y más aún a la predicción de comportamientos subsecuentes o nuevos intereses.

En otro contexto, la minería de datos, y en particular datos en bases de datos espaciales, revelan patrones o asociaciones que usualmente eran desconocidas. Este tipo de bases de datos con cantidades grandes de información (terabytes) son por ejemplo obtenidas de imágenes satelitales, equipos médicos, entre otros (T. Ng y Han 1994).

A estas técnicas se les conoce como Técnicas de Descubrimiento de Conocimiento en Bases de Datos (KDD, por sus siglas en inglés que significan

Knowledge Discovery in Databases) (Holsheimer y Siebes 1994). El descubrimiento de conocimiento en bases de datos es el proceso de identificación de patrones válidos, potencialmente útiles y comprensibles, en los datos (Fayyad 1996). El objetivo es la extracción de conocimiento de los datos especialmente cuando las bases de datos son de gran tamaño. KDD es un proceso multidisciplinario que involucra las áreas de representación del conocimiento, aprendizaje, bases de datos, estadística, sistemas expertos y representación gráfica, entre otros.

Uno de los primeros retos que hace interesante el problema y justifica el estudio de técnicas de reconocimiento de patrones en datos en SSD, es el manejo de grandes cantidades de datos. Para ello, se utilizan técnicas de agrupamiento de datos (*clustering*), las cuales se han aplicado en los últimos 30 años a muchas áreas. En la medicina, por ejemplo, en la clasificación de enfermedades; en la química en el agrupamiento de compuestos equivalentes; en los estudios sociales para la determinación de grupos o redes sociales con base en datos estadísticos, entre otros. El principal problema consiste en identificar estructuras o subclases de los objetos almacenados en las bases de datos espaciales que tengan un significado relevante en su campo de aplicación. La ventaja principal al usar las técnicas de agrupamiento de datos es la posibilidad de descubrir estructuras o categorías bien definidas y relevantes directamente en los datos, sin tener que contar con el conocimiento total del contexto. Ya que el concepto de agrupamiento se define de acuerdo al criterio de similaridad y disimilaridad, y a las diferentes formas (geométricas) de los clusters o patrones que tales criterios inducen, se han desarrollado una gran cantidad de algoritmos. Dichos algoritmos, en términos generales, se pueden clasificar de acuerdo a los tipos de clusters en: esféricos, lineales e irregulares.

A continuación de presentan ejemplos de diferentes tipos de clusters encontrados utilizando un método jerárquico de identificación de clusters conocido como método L (Salvador y Chan s.f.). En la figura 1:2 se presentan 1) clusters del tipo esférico (4000 pts), 2) nueve clusters cuadrados conectados por sus esquinas (9,000 pts), 3) diez clusters esféricos (5,200 pts), 4) Diez clusters

esféricos separados y de diferentes tamaños (5,000 pts), 5) y 6) muestran clustes con diferentes formas (~9,100 pts) y. (~7,600 pts) respectivamente.

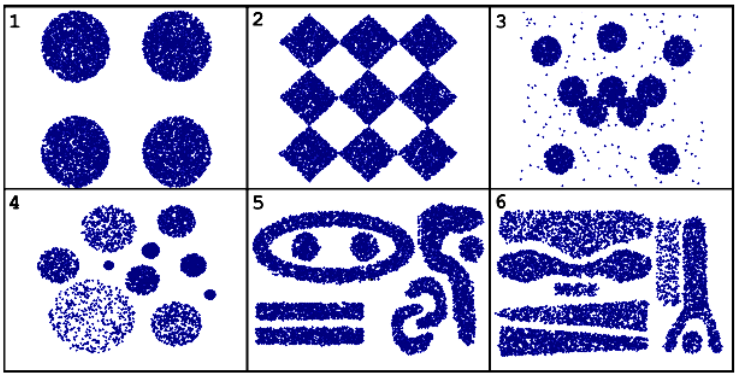


FIGURA 1:2 DIFERENTES TIPOS DE CLUSTERS

Un problema de optimización cuyo correspondiente problema de decisión es NP-completo se le llama problema NP-Hard. Un problema de optimización es aquel que consiste en una función objetivo a optimizar (minimizar o maximizar) y un problema de decisión es aquel que consiste en verificar si la solución al problema de optimización se encuentra dentro de una cota delimitada anteriormente. Un problema NP-completo se puede definir de una manera sencilla como aquel que puede reducirse a otros problemas para los cuales se conjetura que no existe algoritmo eficiente que produzca siempre soluciones óptimas.

Formalmente, el agrupamiento de datos es un problema que consiste en la separación y asignación de datos, definidos como *n*-adas o vectores en espacios multidimensionales, a grupos o clusters, de acuerdo a algún criterio de similaridad. Un *clúster* se define como un conjunto de objetos similares. El criterio de similaridad se establece como la métrica Euclidiana consistente en la función $d:\mathfrak{R}^n\times\mathfrak{R}^n \rightarrow \mathfrak{R}$, que asigna a cualquier par de vectores del espacio Euclidiano n-dimensional $\mathbf{x}=(x_1,x_2,\ldots,x_n)$ y $\mathbf{y}=(y_1,y_2,\ldots,y_n)$, el número $d = \sqrt{\sum_{1\leq i\leq n}(x_i - y_i)^2}$, produciendo así la distancia estándar en el espacio $\mathfrak{R}^n$.

Sin embargo, el problema de agrupar un conjunto de *n* vectores en *k* clusters bajo funciones objetivo es NP-hard, aún cuando los puntos a ser agrupados se restringen al espacio euclidiano bidimensional (Gonzalez 2006).

Así que su complejidad computacional lo hace un problema equivalente, computacionalmente hablando, a aquel bien conocido problema llamado el problema de máximum-cut, el cual es también NP-hard o problema intratable.

Esto nos lleva a concentrarnos en los algoritmos de aproximación eficientes, aunque bajo ciertas condiciones existen heurísticas (formas de trabajo y que apoyan la realización consciente de actividades mentales exigentes) basadas en algoritmos eficientes. Tales algoritmos para resolver el problema de agrupamiento se pueden clasificar de acuerdo a sus métodos representativos como algoritmos basados en: árboles extendidos de costo mínimo, maximum-cut, k-medias (Hartigan 1975), entre otros. Tanto el algoritmo maximum cut como k-medias se explican en el siguiente capítulo.

En esta tesis se aborda un caso de estudio que consiste en analizar la información de una base de datos de 5500 aspirantes a ingresar, en el período de 2003 a 2008, a las diferentes carreras ofrecidas por Facultad de Informática de Universidad Autónoma de Querétaro y que incluye: fecha de nacimiento del aspirante, periodo en el cual hace solicitud para ingresar, escuela de procedencia, resultado oficial obtenido o puntuación, estado de nacimiento, sexo, fecha de nacimiento, si fue o no aceptado y carrera a la que hizo solicitud. A partir de los resultados del análisis de información de los alumnos aspirantes a estudiar en la Facultad de Informática, el departamento de Planeación y Gestión Académica podría tomar mejores decisiones sustentadas en los resultados obtenidos del análisis de la información histórica.

Los algoritmos de agrupamiento se ajustan perfectamente a este tipo de proceso de DM, así como al proceso de agrupamiento (clustering) de datos como una actividad exploratoria. Resulta interesante demostrar que una solución a una instancia del problema de minería de datos, es también una solución al problema de agrupamiento; tanto para una solución factible como óptima (Jain 2000). Este tipo de esquemas de acoplamiento son pasos críticos para lograr las correspondencias semánticas de los atributos a través de fuentes heterogéneas de datos (Yu 2004).

1.2 Procesos de la Minería de Datos

Un proceso típico de DM consta de los pasos generales que se tratan en esta sección.

1.2.1 Recolección de Datos

La recolección de datos es la primera etapa que se requiere previo al diseño de los experimentos. Algunas veces la minería de datos se caracteriza por “el dragado de datos”, esto es tomar conjuntos de datos para dragarlos repetidamente hasta obtener nueva información. Con dragado nos referimos a filtrar aquellos datos que en lugar de ayudarnos en nuestra tarea de exploración e identificación de patrones, podría ser basura. Aunque esta información representa un alto riesgo ya que es recopilada sin una meta específica; también es posible que se pierdan de vista muchos atributos relevantes.

1.2.2 Analizando los tipos de datos

Cuando se está en la fase de recolección de datos es necesario analizar el tipo de datos con los que nos estamos enfrentando. Un dato puede ser tipificado como binario, discreto o continuo. Un dato de tipo discreto tiene un número finito de posibles valores, por lo que los binarios son un caso especial de los discretos (Figura 1:3)

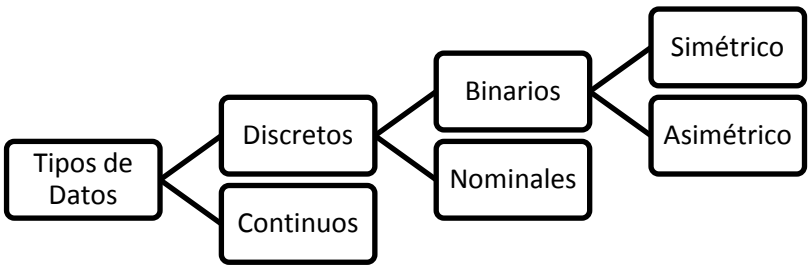


FIGURA 1:3 TIPOS DE DATOS

Para propósitos del presente trabajo se están utilizando datos del tipo discreto, en el primer caso de estudio y del tipo binario en ambos casos, con mayor importancia en el segundo caso de estudio.

1.2.2.1 Datos de tipo Binario

Un dato de tipo binario es aquel que tiene solo dos posibilidades como falso o verdadero, o bien, femenino o masculino.

Un vector binario x con d dimensiones es definido como $(x_1,x_2,\ldots,x_d)$ donde $x_i \in \{0,1\}$ $1 \leq i \leq d$

Se considera importante comentar que existen otros tipos de datos, quizás el tipo de comienza a tomar mayor importancia por sus diversas aplicaciones es el de datos de tipo imagen.

1.2.3 Transformación de los Datos de Entrada

Ningún conjunto de datos es perfecto. Es común encontrar errores en los datos, atributos faltantes, errores de captura, o bien datos duplicados. Batallar con este tipo de problemas es un punto importante a considerar en el proceso de minería de datos, ya que la ausencia de datos precisos lleva a resultados desastrosos.

Existen diversas formas de limpiar los datos, para propósitos del presente trabajo se estará considerando como parte importante de la limpieza de los datos, la conversión de escalas debido a que consideramos datos de diferentes tipos a lo cual llamamos normalizar.

Las escalas de datos tienen una importancia significativa para lograr el clustering. Pueden ser divididas en cuatro escalas cuantitativas y cualitativas. Las escalas cualitativas incluyen escalas nominales y ordinales. Las escalas cuantitativas incluyen las escalas de intervalos y radio, en la figura 1:4 de representan (Jain, M.N.Murty y Flynn 1999)

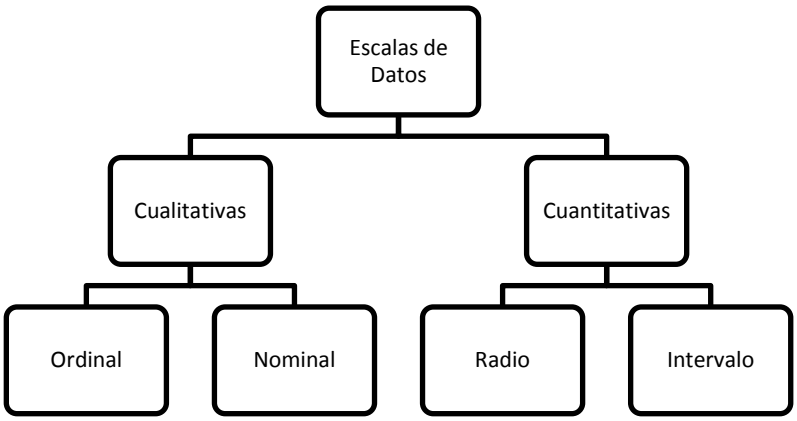


FIGURA 1:4 TIPOS DE ESCALAS DE DATOS

1.2.3.1 Ordinal a Intervalo

La conversión de escala nominal a intervalo crea categorías contiguas sobre la escala del intervalo para cada objeto. Existen diversas formas de lograr la conversión, algunas se apoyan de sofisticadas técnicas estadísticas. Entre las que se utilizan en los casos experimentales que competen al presente trabajo están las siguientes.

a) No hay cambios significativos: por ejemplo cuando se están escalando las variables de edad o escolaridad, que son de tipo radio. Si las utilizamos en términos de años ya no sería necesaria una conversión, ya que naturalmente se estarán ordenando.
b) Sustitución, este método nos indica que se puede reemplazar una variable por otra que esté relacionada y tenga una escala más adecuada a las necesidades del problema. Por ejemplo, el utilizar el nombre de las escuelas de procedencia nos dificulta el análisis de clustering, por lo que se remplaza por la clave asignada a la escuela.

Para una variable de tipo numérico es conocido el intervalo al que pertenece. En el caso de una variable categórica la escala es nominal.

1.2.4 Seleccionar y Aplicar la Técnica de DM

Se construye el modelo predictivo, de clasificación o segmentación.

1.2.5 Extracción del Conocimiento

Mediante una técnica de minería de datos, se obtiene un modelo de conocimiento, que representa patrones de comportamiento observados en los valores de las variables del problema o relaciones de asociación entre dichas variables. También pueden usarse varias técnicas a la vez para generar distintos modelos, aunque generalmente cada técnica obliga a una preparación diferente de los datos.

1.2.6 Interpretando y Evaluando los Resultados

Una vez obtenido el modelo, se debe proceder a su validación comprobando que las conclusiones que arroja son válidas y suficientemente satisfactorias. En el caso de haber obtenido varios modelos mediante el uso de distintas técnicas, se deben comparar los modelos en busca de aquel que se ajuste mejor al problema. Si el modelo final no superara esta evaluación el proceso se podría repetir desde el principio.

Definitivamente la interpretación de los resultados es una de las etapas más complejas debido a que ésta requiere de un conocimiento profundo de los datos que se están manipulando, así como de las reglas de negocio y/o políticas internas de la organización, institución, industria o empresa de la cual se está analizando la información.

Una vez validado el modelo, si resulta ser aceptable (proporciona salidas adecuadas y/o con márgenes de error admisibles) éste ya está listo para su explotación. Los modelos obtenidos por técnicas de DM se aplican incorporándolos en los sistemas de análisis de información de las organizaciones BI, e incluso, en los sistemas transaccionales.

1.2 Clustering

La agrupación de datos (clustering), también conocida como análisis de grupos, análisis de segmentación, análisis de taxonomía, o clasificación sin supervisión, es un método para la construcción de clusters o grupos de objetos, de modo tal que cada par de objetos en un cluster son similares y cada par de objetos, cada uno en diferente cluster son disimilares. El clustering de datos es frecuentemente se confunde con la clasificación donde los objetos son asignados a clases predefinidas. *En el clustering, las clases son creadas al mismo tiempo que los objetos son clasificados*. (Gan, Chaoqun y Jianhong 2007)

Formalmente el problema de clustering consiste en clasificar un conjunto de objetos en grupos homogéneos. Matemáticamente, un grupo de datos dados en un conjunto D puede ser representado por la función:

$f: D \rightarrow [0,1]^k, x \rightarrow f(x)$ definida como sigue

$$f(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_k(x) \end{pmatrix} \quad \ldots . (1)$$

$$donde\ f_i(x) \in [0,1] \quad para\ i = 1,2,3,\ldots,k\ y\ x \in D\ y$$

$$\sum_{i=1}^{k} f_i(x) = 1 \ \forall\ x \ \in D \ \ (2)$$

El clustering constituye una componente importante de los procesos de minería de datos, y consiste en un proceso de exploración y análisis de grandes cantidades de datos con el fin de descubrir información útil.

Clustering está situado como una técnica de DM indirecta donde la minería se hace sin tener un objetivo definido y la meta es descubrir algunas relaciones entre todas las variables, mientras en la minería directa, algunas variables son los objetivos de salida. En clustering de los datos no estamos seguros de lo que se obtendrá. (Gan, Chaoqun y Jianhong 2007). En la figura 1:5 se presentan las tareas asociadas a los dos tipos de DM.

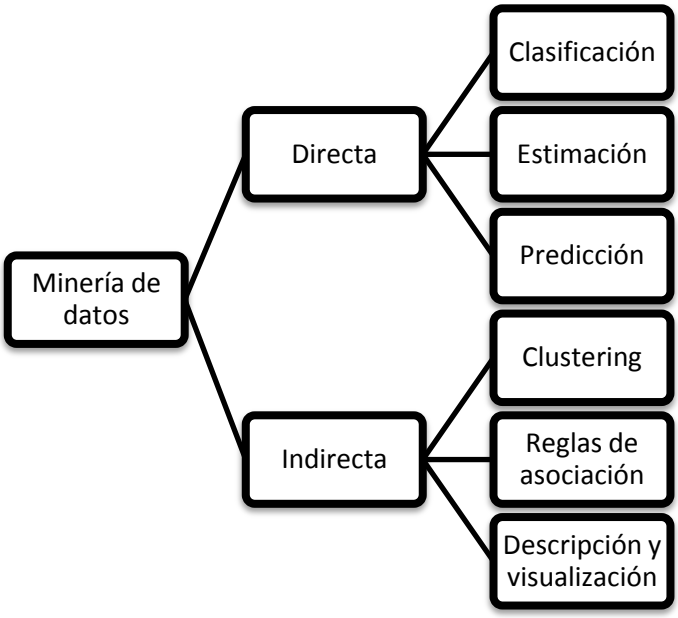


FIGURA 1:5 TIPOS DE MINERÍA DE DATOS

1.2.1 Definiciones

Se introducen algunos conceptos que son comunes en el análisis de clustering.

1.2.1.1 Atributos

Matemáticamente, un conjunto de datos con n objetos donde cada uno está descrito por *d* atributos denotados por $D= \{x_{1,} x_{2,} \dots ,x_{d}\}$ donde $x_i = \{ x_{i1}, x_{i2}, \dots,x_{id} \}$ es un vector denotando el i-ésimo objeto y x_{ij} es una denotación escalar del *j*-ésimo atributo o componente de x_i . El número de atributos d es también conocido como la dimensión del conjunto de datos.

1.2.1.2 Similitudes

Las similitudes juegan un rol importante en el agrupamiento de datos. Las medidas de similitud y sus coeficientes son utilizados para describir cuantitativamente las similitudes entre dos objetos y para descubrir el nivel de disimilaridad entre dos clusters, respectivamente. Así, entre mayor sea la distancia entre los .clusters se tiene que para cada par de datos, cada dato

perteneciendo a clusters diferentes, la medida de disimilaridad describe tanto dichos datos con menos similitudes

Considere dos conjuntos de puntos x = {x_1, x_2, …, x_d} y y = {y_1, y_2, …, y_d}. La distancia Euclideana entre los puntos ***x*** y ***y*** está dada por:

$$d(x,y) = \left(\sum_{j=1}^{d} (x_j - y_j)^2 \right)^{\frac{1}{2}} \text{.....(3)}$$

1.2.1.3 Clusters, Centros y Medias

Clusters, clases, categorías y grupos son utilizados de manera indistinta para hablar de elementos que comparten atributos. El análisis de clusters es una técnica utilizada para la clasificación de los datos. Los elementos de datos se dividen en grupos llamados clusters (se representan en el presente trabajo con la letra k) representan colecciones de elementos de datos y cuya distancia entre ellos es corta. La función de distancia mide la desigualdad de datos entre los diferentes elementos. Un par de elementos idénticos tienen distancia cero y todos los demás tienen una distancia positiva.

Para datos numéricos se sugiere que hay dos tipos de clusters: los compactos y los encadenados. Los del tipo compacto son aquellos en los que los elementos tienen alto índice de similitud entre sí. Normalmente, un cluster compacto se puede representar por un punto representativo o centro. En la figura 1:6 se presentan tres clusters k=3 delimitados con tres colores diferentes.

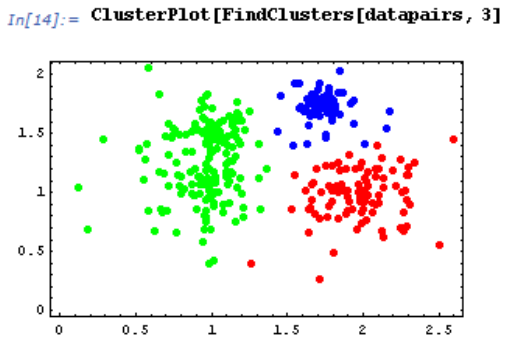


FIGURA 1:6 CLUSTERS COMPACTOS

Los clusters encadenados, son conjuntos de puntos donde cada miembro es más parecido a cada miembro del cluster que aquellos que están fuera de él. Intuitivamente, cualquiera par de puntos de un cluster encadenado son localizables a través de una ruta, es decir, existe por lo menos una ruta que conecte a dos puntos dentro del cluster (Gan, Chaoqun y Jianhong 2007). En la figura 1:7 se observan cinco clusters k=5, se consideran encadenados debido a que comparten elementos.

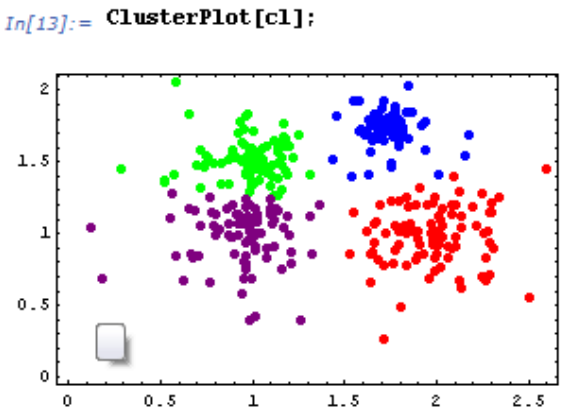


FIGURA 1:7 CLUSTERS ENCADENADOS

1.3 Reconocimiento de Patrones

Un patrón x es un dato utilizado en un algoritmo de clustering, típicamente consiste en un vector de d medidas: $x= (x_1, x_2, .., x_d)$ donde *d* es la dimensión del patrón.

Un conjunto de patrones es denotado por y=$\{y_1, y_2, \ldots, y_n\}$ el i-ésimo patrón de x es denotado por $x_i = (x_{i1}, x_{i2}, .. ,x_{id})$. En muchos casos un conjunto de patrones es representado como una matriz *n x d* (Jain, M.N.Murty y Flynn 1999).

El reconocimiento de patrones se puede definir como la búsqueda de estructuras en los datos (Bezdek y Pal 1992). Esta definición tiene dos implicaciones directas:

1) Es un proceso necesario en muchas líneas de investigación científica.

2) Es una ciencia inexacta, ya que puede admitir aproximaciones para llegar a una solución a un problema dado.

Entre las áreas de aplicación del reconocimiento de patrones se encuentran: en comunicación hombre-máquina la detección de voz automática, el tratamiento de imágenes, el procesamiento de lenguaje natural. En la medicina los diagnósticos, el análisis de imágenes y la clasificación de enfermedades. En aplicaciones policiacas la detección de escritura, de huellas digitales y el análisis de fotografías. En la Industria el diseño asistido por computadora, las pruebas y el control de calidad.

El reconocimiento de patrones consta de varias actividades. Estas son:

1) Elegir el formato de la información, determinando las características que representen los datos del proceso, y así filtrar las no significativas.
2) Agrupar los datos caracterizados, etiquetando los subgrupos naturales y homogéneos que se encuentren en el espacio *n*-dimensional de acuerdo a las *n* características establecidas.
3) Por último, diseñar un clasificador, capaz de etiquetar cualquier punto del espacio de características, esto es, asociar a cada dato el cluster al que pertenece.

En el caso del presente trabajo el reconocimiento de patrones toma un papel importante debido a ser parte de las técnicas de clustering que se estarán utilizando en los casos de estudio.

1.4 Normalización de los Datos

En el presente trabajo se estará llamando normalización de datos al ajuste de la escala de datos que se realiza. La cual es necesaria para algunos datos, de otra forma la escala que se utiliza sería demasiado amplia. Un ejemplo que caracteriza esta situación es la de las claves de las escuelas, las cuales van de un rango desde 1 hasta 700, pero no de todas las escuelas de procedencia se

han tenido aspirantes a ingresar, por lo que se registran claves variadas y discontinuas.

Un punto muy importante en cualquier problema de agrupamiento, y que se debe estudiar antes de ejecutar cualquier algoritmo, es el decidir si se debe aplicar algún tipo de normalización a los datos. Esto se debe a que las fórmulas para calcular las distancias son sensibles a las variaciones en los rangos de las variables que representan las características. En particular, la distancia Euclidiana da más peso a las variables que tienen rangos más amplios que a las que tienen rangos más estrechos. De esta forma, es aconsejable aplicar algún tipo de normalización antes de ejecutar la técnica de clustering que sea seleccionada (Jai, 1988). Hay varios tipos de normalización que pueden aplicarse, siendo los siguientes los más habituales:

- Restar a cada valor de la característica la media de los valores de dicha característica.
- Establecer una transformación lineal de modo que el rango corresponda al intervalo cerrado [0,1].

Es importante establecer que, en algunos contextos, la normalización no siempre es deseable, ya que se puede alterar la separación entre conjuntos e influenciar negativamente los resultados del agrupamiento. Retomando el ejemplo da las claves de escuelas de procedencia, es deseable que las claves estén asignadas de manera consecutiva para evitar que los clusters se vean afectados.

No obstante, los algoritmos capaces de adoptar las formulas para calcular la distancia son menos sensibles al escalamiento de los datos, ya que pueden compensar estas diferencias.

1.5 Hipótesis

El problema de DM no es nuevo se remonta a las técnicas de clustering, en este sentido en la presente tesis se considera el utilizar dichas técnicas para apoyar a la toma de decisiones, en especifico se utilizan casos de estudios aplicados a una institución educativa. Para lo que se presenta la siguiente hipótesis:

El uso de algoritmos para resolver el problema de agrupamiento (clustering) es apropiado para efectuar la actividad exploratoria propia del proceso de minería de datos, aplicado a nuestro caso de estudio referente a datos académicos. La hipótesis será probada mediante la validación del método, haciendo uso de datos históricos y comparando los resultados con las decisiones hechas en el pasado.

1.6 Método de Investigación

El método de investigación a seguir es el hipotético-deductivo en el que la hipótesis planteada se analizará deductiva o inductivamente y posteriormente se comprobara experimentalmente. Concretamente el método abordara los siguientes puntos relevantes:

1) Investigación y selección de algoritmos apropiados para el análisis de la información.
2) Evaluación experimental de los algoritmos.
3) Modelación de la información de acuerdo a los datos provenientes de las bases de datos.
4) Modelado del prototipo en diferentes herramientas.
5) Análisis de los resultados.
6) Elaboración de conclusiones.

1.7 Estructura de la Tesis

En este proyecto de tesis se contemplan cinco fases importantes.

Primeramente, el capítulo uno trata de dar una introducción a los temas que se estarán abordando durante el presente trabajo.

En el capítulo dos se mencionan algunos de los algoritmos de clustering que se estarán utilizando basados en heurísticas tradicionales.

En el tercer capítulo se procede a la implementación de aquellos algoritmos relevantes usando el lenguaje Mathematica. Dicho lenguaje será utilizado por sus ventajas gráficas y cualidades de lenguaje basado en la programación funcional, que facilitan un rápido y eficiente proceso de construcción de prototipos. Se utiliza un caso de estudio utilizando información de los aspirantes a la Facultad de Informática.

En el capítulo cuatro se hace la herramienta de Análisis de Información propia de SQL Server 2008, se utiliza el mismo caso de estudio que en el capitulo anterior.

En el capítulo cinco se modela un problema de Agrupamiento de Entradas Binarias, utilizando instancias sintéticas de vectores para la prueba de los algoritmos. Se presenta la aplicación de clustering y su relación con problemas de optimización para encontrar una mejor solución.

Finalmente, procederé a la elaboración de conclusiones.

Es importante mencionar que en el presente trabajo se utilizan de manera indistinta tanto la palabra clustering como agrupamientos.

1.8 Base de Datos Utilizada para los Casos de Estudio

Antes de comenzar a explicar la base de datos que se utilizó es necesario comentar que los alumnos aspirantes a entrar a la Facultad de Informática presentan el Examen de Habilidades y Conocimientos Básicos (EXCOBA, http://iide.ens.uabc.mx/exhcproc.html) el cual es el examen institucional desde el año 2005. La puntuación con la que se aceptan a los alumnos es responsabilidad de cada Facultad, y de cada dirección.

La base de datos utilizada en los diferentes casos de estudio que se presentan en este trabajo, consiste en una Tabla que incluye 19 registros. De esta Tabla se seleccionaron aquellos parámetros que potencialmente

representan un alto factor de correlación. Los datos seleccionados son los indicados en la Tabla 1-1.

TABLA 1-1 SELECCIÓN DE ATRIBUTOS DE LA BASE DE DATOS PARA LOS CASOS DE ESTUDIO

Atributo	Descripción
SEXO	Femenino o masculino.
CVECARR	Clave correspondiente a la carrera que aspira el alumno.
RESUOFICIA	Resultado obtenido en el examen de admisión.
CVEESCORI	Clave de la escuela de procedencia del aspirante.
EDONAC	Estado de nacimiento del aspirante
FECNAC	Fecha de nacimiento del aspirante.

La cantidad de datos que se está manipulando es de un poco más de 5000 registros. De los cuales se estarán utilizando con diversos filtros para así, implementar el análisis de la información. Con filtros significan que durante el análisis de la información se detectó la necesidad de normalizar los datos. En la siguiente sección se presentan dichas normalizaciones.

1.8.1 Normalización de los datos

La normalización de los datos se utilizará en la herramienta Mathematica, en el caso de la herramienta SQL no se realizará ya que debido a las características de ésta última herramienta permite visualizar la información sin necesidad de hacerlo.

1.8.1.1 Sexo

En el caso de sexo la normalización es muy simple. Se muestra en la Tabla 1-2.

TABLA 1-2 NORMALIZACIÓN DE LAS CLAVES DE SEXO

Sexo	**Clave**
Masculino	1
Femenino	0

1.8.1.2 Planes de Carrera

La Tabla 3 presenta las claves asociadas a las carreras que se imparten en la Facultad de Informática. Esta información se estará analizando en los análisis experimentales que se presentan más abajo.

TABLA 1-3 CLAVES DE CARRERAS A LOS QUE PUEDEN ASPIRAR CON COLUMNA NORMALIZADA

Clave Carrera	**Nombre de la Carrera**	**Clave del plan**	**Clave normalizada**
19	INFORMATICA	INF07	1
22	INGENIERIA EN COMPUTACION	INC99	2
30	TECNICO SUPERIOR EN COMPUTACION Y REDES	TSC02	3
47	INGENIERIA DE SOFTWARE	SOF07	4
48	INGENIERIA EN TELECOMUNICACIONES	TEL07	5

1.8.1.3 Resultados oficiales (RESUOFICIA) o Calificaciones

El rango de los resultados oficiales va desde 0 hasta 177.25. No requieren normalización.

1.8.1.4 Estados de Nacimiento

Es importante mencionar que los estados de nacimiento de los alumnos no refieren estados donde radican ni estados donde hicieron sus últimos estudios, sin embargo se consideró interesante utilizar para los experimentos. En el caso de los estados la normalización no se aplica. Sin embargo se muestra la Tabla con los estados y las claves asociadas.

TABLA 1-4 RELACIÓN DE ESTADOS Y CLAVES

Clave	Estado	Clave	Estado
1	AGUASCALIENTES	18	NAYARIT
2	BAJA CALIFORNIA SUR	19	NUEVO LEÓN
5	COAHUILA	20	OAXACA
7	CHIAPAS	21	PUEBLA
8	CHIHUAHUA	22	QUERÉTARO
9	DISTRITO FEDERAL	24	SAN LUIS POTOSÍ
10	DURANGO	25	SINALOA
11	GUANAJUATO	26	SONORA
12	GUERRERO	28	TAMAULIPAS
13	HIDALGO	29	TLAXCALA
14	JALISCO	30	VERACRUZ
15	ESTADO DE MÉXICO	31	YUCATÁN
16	MICHOACÁN	32	ZACATECAS
17	MORELOS	33	EXTRANJEROS

Nótese que solo se encuentran los estados de los cuales se han tenido aspirantes.

1.8.1.5 Claves de Escuelas

Las claves de las escuelas de las que provienen los aspirantes son muy diversas debido a como se menciona arriba que se tienen aspirantes de diferentes partes de la República mexicana.

El rango que comprende va desde 1 a 625.

{1, 2, 3, 4, 5, 7, 8, 11, 13, 14, 15, 16, 20, 21, 22, 23, 24, 25, 26, 27, 29, 30, 31, 34, 36, 37, 38, 39, 40, 41, 46, 47, 48, 51, 53, 62, 63, 64, 67, 68, 70, 72, 74, 75, 77, 78, 80, 81, 83, 84, 85, 86, 87, 88, 90, 94, 95, 98, 99, 104, 105, 106, 107, 108, 110, 112, 114, 116, 128, 130, 131, 132, 133, 137, 139, 312, 314, 348, 349, 402, 408, 411, 413, 417, 500, 501, 511, 517, 519, 608, 620, 624, 625}

Apoyándonos de la función **Map** de Mathematica, se hace el mapeo entre la clave de la escuela y el valor consecutivo asociado.

$$\text{changes} = \text{Map}[\#_{[[1]]} \rightarrow \#_{[[2]]}\&, Thread[\{nCarr = Union[Map[\#_{[[2]]}\&, lista]], Range[Length[nCarr]]\}]]$$

{1→1,2→2,3→3,4→4,5→5,7→6,8→7,9→8,11→9,13→10,14→11,15→12,16→13,19→14,20→15,21→16,22→17,23→18,24→19,25→20,26→21,27→22,29→23,30→24,31→25,32→26,34→27,36→28,37→29,38→30,39→31,40→32,41→33,46→34,47→35,48→36,51→37,52→38,53→39,54→40,62→41,63→42,64→43,67→44,68→45,70→46,72→47,74→48,75→49,77→50,78→51,80→52,81→53,83→54,84→55,85→56,86→57,87→58,88→59,90→60,91→61,94→62,95→63,98→64,99→65,104→66,105→67,106→68,107→69,108→70,110→71,112→72,114→73,116→74,124→75,128→76,130→77,131→78,132→79,133→80,135→81,136→82,137→83,139→84,312→85,314→86,348→87,349→88,402→89,407→90,408→91,409→92,411→93,412→94,413→95,415→96,416→97,417→98,419→99,420→100,424→101,425→102,428→103,429→104,430→105,432→106,500→107,501→108,502→109,509→110,511→111,513→112,515→113,516→114,517→115,519→116,521→117,530→118,603→119,608→120,612→121,617→122,620→123,624→124,625→125}

Finalmente las claves de las escuelas quedan de la siguiente manera:

{1, 1}, {2, 2}, {3, 3}, {4, 4}, {5, 5}, {7, 6}, {8, 7}, {11, 8}, {13, 9}, {14, 10}, {15, 11}, {16, 12}, {20, 13}, {21, 14}, {22, 15}, {23, 16}, {24, 17}, {25, 18}, {26, 19}, {27, 20}, {29, 21}, {30, 22}, {31, 23}, {34, 24}, {36, 25}, {37, 26}, {38, 27}, {39, 28}, {40, 29}, {41, 30}, {46, 31}, {47, 32}, {48, 33}, {51, 34}, {53, 35}, {62, 36}, {63, 37}, {64, 38}, {67, 39}, {68, 40}, {70, 41}, {72, 42}, {74, 43}, {75, 44}, {77, 45}, {78, 46}, {80, 47}, {81, 48}, {83, 49}, {84, 50}, {85, 51}, {86, 52}, {87, 53}, {88, 54}, {90, 55}, {94, 56}, {95, 57}, {98, 58}, {99, 59}, {104, 60}, {105, 61}, {106, 62}, {107, 63}, {108, 64}, {110, 65}, {112, 66}, {114, 67}, {116, 68}, {128, 69}, {130, 70}, {131, 71}, {132, 72}, {133, 73}, {137, 74}, {139, 75}, {312, 76}, {314, 77}, {348, 78}, {349, 79}, {402, 80}, {408, 81}, {411, 82}, {413, 83}, {417, 84}, {500, 85}, {501, 86}, {511, 87}, {517, 88}, {519, 89}, {608, 90}, {620, 91}, {624, 92}, {625, 93}}

Esto quiere decir que el rango ahora es continúo desde escuela con clave 1 a 93. Por ejemplo, revisando el vector que se presenta arriba la clave 625 es remplazada por la clave 93. Y así es para todos los casos.

Los nombres de las escuelas registradas se presentan en el Apéndice, debido a la extensión.

Como se puede ver es significativa la reducción de claves y nos permitirá ver de manera más clara los clusters cuando se apliquen en algún experimento.

1.8.1.6 Fechas de nacimiento

En cuanto a las fechas de nacimiento de los aspirantes a la Facultad de Informática, tienen un rango 1963 a 1991. La normalización consiste en obtener una edad aproximada utilizando el periodo en el cual el aspirante fue aceptado y su fecha de nacimiento. Los periodos comprendidos son del 2003 al 2008 recordando que cada año tiene dos periodos en los que los alumnos ingresan. Por ejemplo en el 2003-1 es el periodo que comprende desde enero de 2003

hasta julio 2003 y a partir de agosto y hasta diciembre del 2003 se considera 2003-2.

Al normalizar las fechas de nacimiento se obtiene un rango de edades que comprende desde 17 años hasta 41 años.

2 Algoritmos de Agrupamiento (clustering)

LAS MATEMÁTICAS SON EL ALFABETO CON EL CUAL DIOS HA ESCRITO EL UNIVERSO. GALILEO GALILEI

Como un problema fundamental de reconocimiento de patrones, un algoritmo de clustering bien diseñado, generalmente involucra las siguientes fases de diseño: representación de los datos, modelado, optimización y validación.

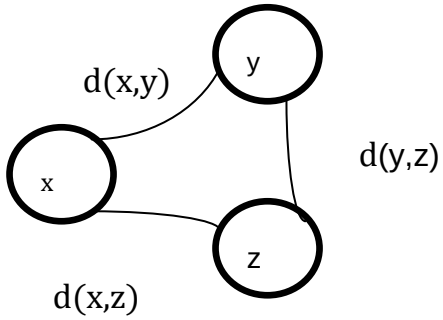
Los algoritmos de agrupamiento llevan a cabo la clasificación de datos basados en criterios de similitud entre objetos. Los grupos o clusters contienen datos de modo que los datos en un cluster difieren de los datos ubicados en otros grupos. Desde la perspectiva computacional, los clusters obedecen a patrones ocultos. Los algoritmos de clustering se utilizan en problemas de minería de datos cuando se está trabajando con cantidades grandes de datos, muchas variables y atributos de diferentes tipos. Dichas características convierten a la base de datos en una entidad compleja de analizar y minar.

Existen una gran variedad de técnicas para representar datos, así como medidas de proximidad (similitudes) entre los datos. Formalmente, asumimos que la entrada es un conjunto de datos en un espacio métrico, con una función de distancia asociada. Denotaremos esta distancia como d, entonces la distancia entre dos puntos x y y es dada por $d(x,y)$. Si los puntos están en un espacio métrico entonces se tienen tres requerimientos:

1. **Identidad**: $d(x,x) = 0$; esto es la distancia de cualquier punto así mismo es cero.

2. **Simetría**: $d(x,y) = d(y,x) > 0$ esto es, la distancia entre uno o dos puntos es la misma en ambas direcciones y es no negativa.

3. **Desigualdad Triangular**: $d(x,z) \leq d(y,x) + d(y,z)$; esto es, la longitud entre los puntos jamás superará la longitud total entre dos puntos a través de una ruta que incluye un tercer punto intermedio. (Ver Figura 2:1)



2.1 Definiciones

2.1.1 Clusters

Definición 1 *Un algoritmo de clustering toma como entrada un conjunto de puntos en un espacio multidimensional métrico y como resultado un conjunto de grupos (clusters),* $C = \{ C_1 \ldots C_k \}$.

Esta definición es extensa, no describe como se forman los clusters o que criterio se considera para crearlos. Esto es porque hay diferentes formas o criterios para la definición de los clusters.

2.1.2 k-center

Definición 2 *El algoritmo de clustering k-center también conocido como el problema de k-center (kCP), genera un conjunto de k puntos* $C = \{c_1 \ldots c_k\}$ *(centros). Cada dato de entrada es asociado con un punto en C que sea más cercano a él. La calidad de los grupos está determinada por la distancia máxima de un punto a su centro más cercano, es decir,*

$$max_x \left\{ min_i \{ d(x, C_i) \} \right\} \quad (4)$$ (B.Xiao 2001)

2.1.2.1 Pseudocódigo KCP

1: Flag=No
2: Para i =1,…,$\binom{n}{p}$
3: Seleccione p puntos diferentes de C que genera un nuevo conjunto de puntos T_i con $T_i \neq T_j$ (j = 1, 2 ,… , i-1)
4: Seleccione un punto de T_i, c_1=t, C={c_1}
5: Para j=2,…,k, hacer
6: Para un punto t ∈ T_i y t ∉C entonces d_j(t)=min[distancia(t, c_j), Para ∀ $c_j \in C$
7: Hacer $d_j = \max t \in T_i\ d_j$ (t)
8: Hacer c_j sea el punto t que hace que d_j sea el valor máximo
9: C=C ∪ {c_j}
10: Si $d_k \leq 4r$
11: Si k discos con centros **c** y con radio **r** pueden cubrir al menos p puntos s en S
12: Flag = SI
13: Si Flag= NO, es imposible cubrir **p** puntos con **k** discos
14: Regresa NO

2.1.2.2 Ejemplo

Suponga que se tienen k puntos distribuidos aleatoriamente en un plano de dos dimensiones, el problema es encontrar los clusters que agrupen por características similares a los k puntos. ¿Es posible que K discos o clusters agrupen k puntos? En la Figura 2:1 (a) se tienen k puntos los cuales se agruparan en cinco clusters. Lo primero es seleccionar los centros c, los cuales se consideran parte de un conjunto C. En la Figura 2:1 (b), se comienzan a medir las distancias entre los centros. Comparando con todos los centros para encontrar la distancia mínima y así seleccionar los centros como parte de los clusters (vea la Figura 2:1 (c)). El algoritmo continúa recursivamente desde 1 hasta $\binom{n}{p}$, agrupando los k puntos en n clusters, minimizando la distancia entre los puntos y el centro c. Y tratando de cubrir todos los puntos.

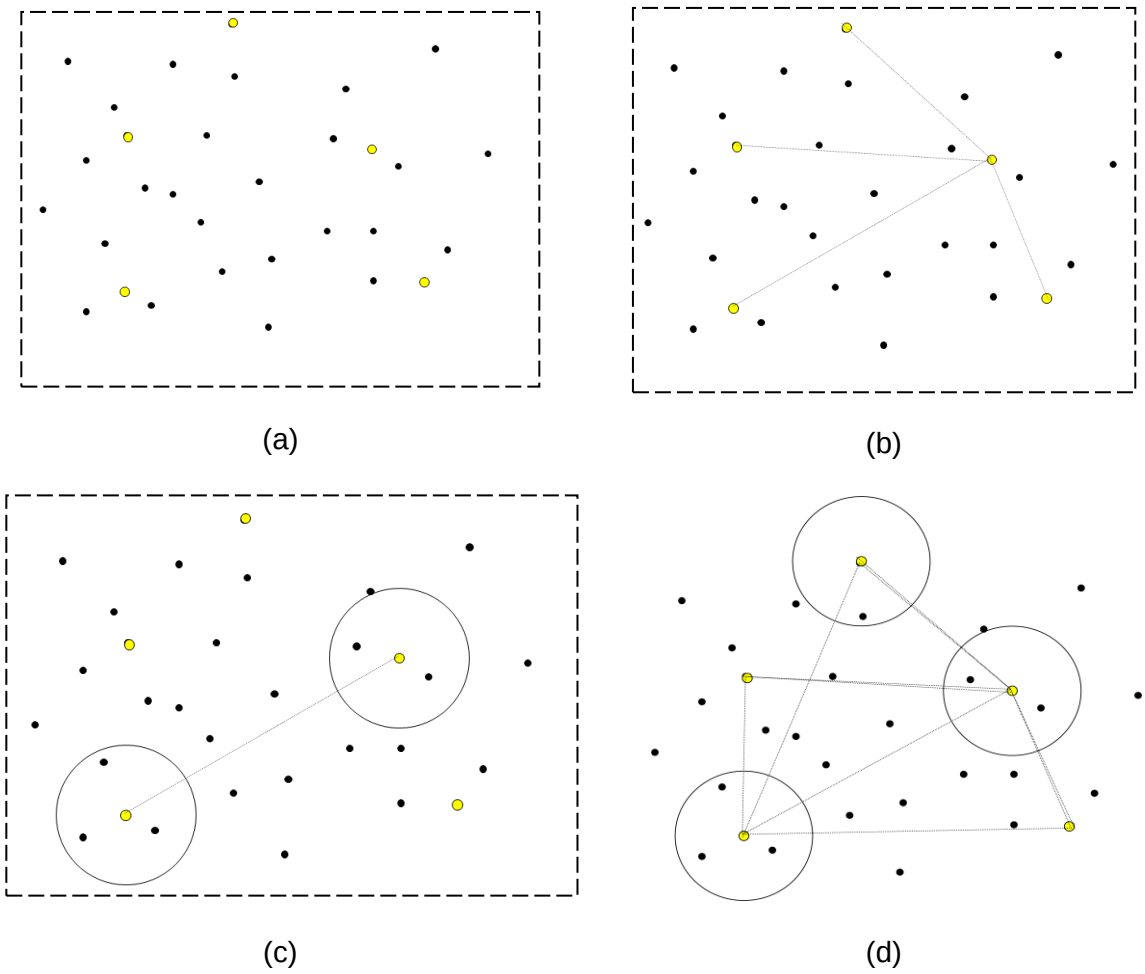


FIGURA 2:1 EJEMPLO DE KCP

2.1.3 k-media

Definición 3 *El algoritmo de clustering k-media da como resultado un conjunto C de k puntos (medias) en un espacio métrico que se definen los clusters: cada punto de entrada es asociado con el punto c que están muy cerca de él. La calidad de los clusters está determinada por el promedio de la distancia de los puntos más cercanos al centro, esto es* $\frac{1}{n}\sum x\,min_i\ d(x,C_i)$

Nótese que en ambas definiciones la manera en que se ubican los clusters es la misma, lo que cambia es la función objetivo (ya sea maximizar o minimizar).

Mas técnicamente, se sabe que el problema de clustering para las funciones objetivos establecidas en las definiciones 2 y 3 son problemas de optimización que pertenecen a la clase de problemas llamados NP-duros (Jhonson y Garey).

2.1.4 Distancias entre Clusters

Las distancias entre los clusters son funciones de las distancias entre datos, y hay varias formas de definirlas: Sean A y B dos clusters.

1) Vecino más cercano: d(A, B) = mínimo {d (i, j)} donde i ∈ A, j ∈ B. El vecino más cercano tiende a formar clusters más alargados.
2) Vecino más lejano: d(A, B) = máximo {d(i, j)} donde i ∈ A, j ∈ B. El vecino más lejano forma clusters más esféricos.
3) Centroide: d(A, B) = d($\bar{x}_A$, $\bar{x}_B$) en qué $\bar{x}_A$ y $\bar{x}_B$ son los respectivos centroides de los conglomerados A y B.
4) Medoides: es la distancia entre los medoides de los grupos. Es el punto tal que sus coordenadas son las medianas de las variables respectivas.

En la Figura 2:2 se presenta gráficamente las diferentes maneras de calcular las distancias para definir los clusters.

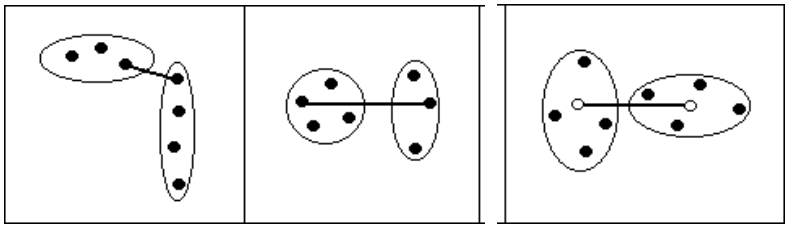


FIGURA 2:2: DISTANCIAS ENTRE CONGLOMERADOS: VECINO MÁS CERCANO, MÁS LEJANO, CENTROIDE

2.1.5 Categorias de Algoritmos de Clustering

Los algoritmos de clustering se agrupan en dos categorías (Figura 2:3):

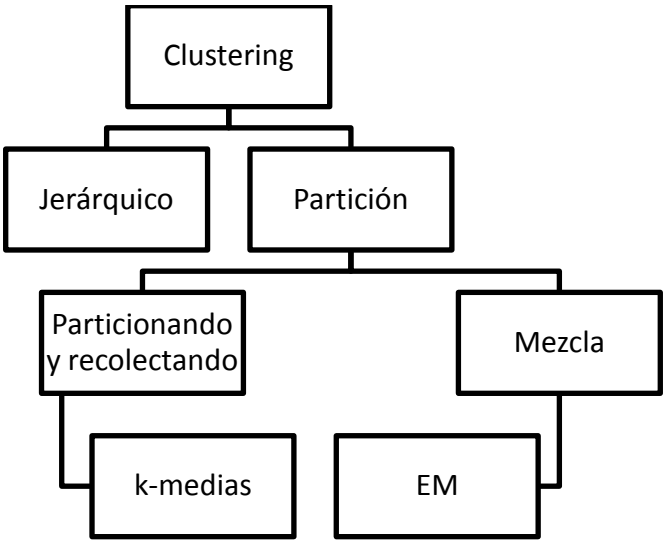


FIGURA 2:3 TIPOS DE ALGORITMOS DE CLUSTERING

Algoritmos jerárquicos: Estos se caracterizan por generar una estructura de árbol (llamada dendograma, Figura 2:4), en la que cada nivel es un agrupamiento posible de los objetos de la colección [Jain *et.al.*, 1999; Dash *et.al.*, 2001; Han, 2001]. Cada vértice (nodo) del árbol es un grupo de elementos. La raíz del árbol (primer nivel) se compone de un sólo grupo que contiene todos los elementos. Cada hoja del último nivel del árbol es un grupo compuesto por un sólo elemento (hay tantas hojas como objetos tenga la colección). En los niveles intermedios, cada nodo del nivel n es dividido para formar sus hijos del nivel $n + 1$.

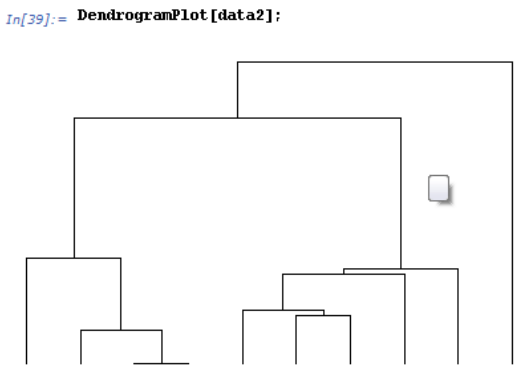


FIGURA 2:4 Ejemplo de un Dendograma

Algoritmos de partición: están basados en métodos que dividen el conjunto de observaciones en k grupos, donde k es dado por el usuario.

Los métodos de partición, a diferencia de los jerárquicos, no van generando distintos niveles de agrupamiento de los objetos, sino que trabajan en un sólo nivel, en el que se refina (optimiza), un cluster. Estos métodos asumen que el valor de k (la cantidad de grupos), está definida de antemano.

2.2 Algoritmo k-medianas

Es como el k-medias, usa los medoides en lugar de los centroides.

Dado un grafo ponderado (es decir que tiene pesos en cada uno de sus aristas) *G = (V, E)*, la meta es agrupar los vértices *v* del grafo *G* en *k* clusters de modo que una determinada función llamada función objetivo sea optimizada.

En k-medianas se divide el conjunto de vértices *v* en *k* clusters, cada uno con un vértice distinguido llamado centro o mediana, de modo que la suma de las distancias desde los vértices a las medianas de sus clusters sea minimizada.

Para encontrar una solución nos apoyamos de algoritmos de búsqueda local. La búsqueda local es la base de muchos de los métodos usados en problemas de optimización. Se puede ver como un proceso iterativo que empieza en una solución y la mejora realizando modificaciones locales. Básicamente empieza con una solución inicial y busca en su vecindad por una mejor solución.

Si la encuentra, reemplaza su solución actual por la nueva y continua con el proceso, hasta que no se pueda mejorar la solución actual.

Otro problema será minimizar la distancia interior del cluster. Teniendo un grafo no dirigido completo ponderado *G = (V, E, W)* la partición de v en B_1, B_2,.. B_k clusters. Para minimizar las distancias entre todos los puntos, maximizaremos las distancias entre los puntos de los diferentes clusters, ya que son equivalentes. Para lo cual nos apoyamos de k-maxcut, por lo que se presenta el pseudocódigo.

2.2.1 Pseudocódigo k-MAXCUT

1: Hacer $S_i = 0$ para toda i
2: Para i = 1 hasta n Hacer
3: Wj = $\sum_{u \in s_j} w(v_i, u)$ para $1 \leq j \leq k$
4: // $W_j = 0$ si no hay aristas desde v_i a un vértice en S_j
5: Asignar v_i a S_l si $W_l \leq W_j \; \forall \, j$ y l causa el costo menor posible
6: Fin ciclo
7: Fin Procedimiento

2.2.2 Ejemplo

Dado el siguiente grafo (Figura 2:5) se pretende buscar dos clusters agrupado de manera que maximice el valor de la función objetivo

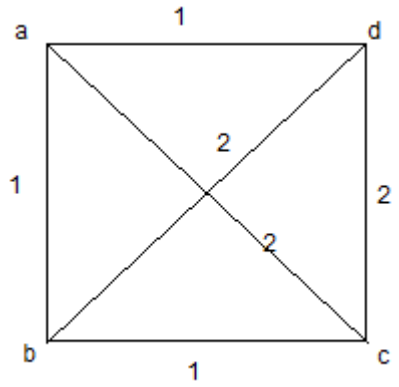


FIGURA 2:5 GRAFO NO DIRIGIDO PONDERADO

Por lo que se comienza con las siguientes consideraciones:

```
1: {
2:      S1 ← V
3:      S2 ← 0
4:      Mientras exista un vértice tal que moviéndolo al otro
        conjunto mejore la solución
5:          Mover el vértice al otro conjunto
6:      Fin
7: }
```

Iteración 1	
S_1	S_2
a	0
b	
c	
d	

⇒

Iteración 2		
S_1	S_2	*Pesos*
a	d	1
b		2
c		2
Función objetivo =		5

⇒

Iteración 3		
S_1	S_2	*Pesos*
a	d	1
a	c	2
b	d	1
b	c	2
Función objetivo =		6

Por lo que la Función objetivo queda maximizada y los clusters quedan identificados: S_1 = *{a,b} y* S_2 = *{d,c}*

2.3 Algoritmo k-medias

El algoritmo de *k*-medias es un método que consiste en asignar cada dato en un solo cluster. Esto significa que un punto representando un dato solo puede pertenecer a un solo clúster.

En *k*-medias el usuario asigna un número de grupos que servirán como clusters. La agrupación en clústeres k-medias es un método muy conocido para asignar la pertenencia al clúster que consiste en minimizar las diferencias entre los elementos de un clúster al tiempo que se maximiza la distancia entre los clusters. El término "media" hace referencia al *centroide* del clúster, que es un punto de datos que se elige arbitrariamente y que se refina de forma iterativa hasta que representa la verdadera media de todos los puntos de datos del clúster. La *k* hace referencia a un número arbitrario de puntos que se utilizan para inicializar el proceso de agrupación en clústeres. El algoritmo k-medias calcula las distancias euclidianas cuadradas entre los registros de datos de un clúster y el vector que representa la media de clústeres, y converge en un conjunto final de *k* clústeres cuando la suma alcanza su valor mínimo.

El algoritmo k-medias asigna cada punto de datos a un solo clúster y no permite la incertidumbre en la pertenencia. En un clúster, la pertenencia se expresa como una distancia desde el centroide.

Normalmente, el algoritmo k-medias se utiliza para crear clusters de atributos continuos, donde el cálculo de la distancia a una media se realiza de manera sencilla.

El algoritmo requiere que:

1. Se indique el número de k clusters a localizar.
2. Dividir el conjunto de datos en k grupos iníciales.
3. Recorrer todos los datos observaciones, asignándolas al cluster cuyo centroide esté a menor distancia. Cada vez que se reasigna una observación a un cluster distinto del que la contenía se deben

volver a calcular los centroides del cluster que pierde la observación y del que la recibe.

4. Si el cluster *A* (que consiste en n_A observaciones) pierde la observación *xi* y si el cluster *B* (con n_B observaciones) recibe a *xi* , los centroides respectivos x_A y x_B se modifican de la siguiente forma:

$$\bar{x}'_{\mathrm{A}} = \frac{1}{n_A - 1}\left(n_A\,\overline{x_A} - x_i\right) \quad (5)$$

$$\bar{x}'_{\mathrm{B}} = \frac{1}{n_B - 1}\left(n_B\,\overline{x_B} - x_i\right) \quad (6)$$

5. Repetir el paso 3 hasta que no haya más reasignaciones.

2.3.1 Ejemplo k-medias

Suponga que se tienen cuatro observaciones cuya matriz de datos está dada a continuación:

$$\begin{bmatrix} 0 & 3 & 9 & 12 \\ 4 & 1 & 6 & 10 \\ 10 & 7 & 3 & 4 \\ 10 & 10 & 3 & 1 \end{bmatrix}$$

Se usará el método de k-medidas para formar dos clusters k=2. Apoyándonos de distancias Euclidianas. En forma de vectores, las cuatro observaciones (filas) son:

$$\bar{x}_1 = \begin{bmatrix} 0 \\ 3 \\ 9 \\ 12 \end{bmatrix} \qquad \bar{x}_2 = \begin{bmatrix} 4 \\ 1 \\ 7 \\ 10 \end{bmatrix} \qquad \bar{x}_3 = \begin{bmatrix} 10 \\ 7 \\ 3 \\ 4 \end{bmatrix} \qquad \bar{x}_4 = \begin{bmatrix} 10 \\ 10 \\ 3 \\ 1 \end{bmatrix}$$

Definimos arbitrariamente dos clusters iníciales:

$$\mathrm{A} = \{\bar{x}_1\} \qquad \mathrm{B} = \{\bar{x}_2, \bar{x}_3, \bar{x}_4\}$$

Sus centroides respectivos son:

$$\bar{x}_A = \begin{bmatrix} 0 \\ 3 \\ 9 \\ 12 \end{bmatrix} \quad \bar{x}_B = \begin{bmatrix} 8 \\ 6 \\ 4 \\ 5 \end{bmatrix}$$

Iteración 1: Cuadro de distancias euclidianas (al cuadrado) de las observaciones a los centroides, partiendo por $\bar{x}_1$.

Observación	**Centroide**	
	$\bar{x}_A$	$\bar{x}_B$
$\bar{x}_1$	0	147
$\bar{x}_2$	33	70

Cambia $\bar{x}_2$ del cluster B a A y termina la iteración 1. No es necesario seguir probando con $\bar{x}_3$ ni $\bar{x}_4$.

Iteración 2: Nuevos centroides: A= $\{\bar{x}_1, \bar{x}_2\}$ B= $\{\bar{x}_3, \bar{x}_4\}$

$$\bar{x}_A = \begin{bmatrix} 2 \\ 2 \\ 7.5 \\ 11 \end{bmatrix} \quad \bar{x}_B = \begin{bmatrix} 10 \\ 8.5 \\ 3 \\ 2.5 \end{bmatrix}$$

Cuadro de distancias al cuadrado, partiendo de $\bar{x}_3$:

Observación	**Centroide**	
	$\bar{x}_A$	$\bar{x}_B$
$\bar{x}_3$	158.25	4.5
$\bar{x}_4$	248.25	4.5
$\bar{x}_1$	8.25	256.5
$\bar{x}_2$	8.25	157.5

Los clusters resultantes son: A= $\{\bar{x}_{1,}\ \bar{x}_2\}$ B= $\{\bar{x}_3,\bar{x}_4\}$

2.3.2 Pseudocódigo

Requiere de entrada un conjunto de ítems x definidos en un espacio Euclidiano, así como el número k deseado de clusters.

```
1:  Para 1 ≤ i ≤ k hacer
2:    kmeans[i] ‹— item aleatorio de los datos
3:    Centroid[i] ‹— 0
4:    Count[i] ‹— 0
5:  repetir
6:  Para all x ∈ ítems hacer
7:     mindist ‹— 0
8:     Para 1  ≤ i ≤ k hacer
9:       Si ‖x – kmeans[i]‖2 < ‖x – kmeans[mindist]‖2
10:        entonces  Mindist ‹— i
11:    Cluster[x] ‹— mindist
12:    Centroid[mindist] ‹— Centroid[mindist] + x
13:    Count[mindist] ‹—  count[mindist] + 1
14:   Para 1 ≤ i ≤ k hacer
15:     Kmeans[i] ‹— Centroid[i]/count[i]
16:     CentRoid[i] ‹— 0
17:     Count[i] ‹— 0
18: Hasta no items reclasificados o excedan count
19: Cada x que pertenecen a ítems ahora están clasificados en cluster[x]
```

K-medias consiste en dividir el conjunto de vértices *v* en *k* clusters, cada uno con un vértice distinguido llamado centro o mediana, de modo que la suma de las distancias desde los vértices a las medianas de sus clusters sean minimizadas.

2.4 Maximización de Expectación (EM)

Otra técnica es el método Expectation Maximization (EM), es un método de agrupación en clústeres blando. Esto significa que un punto de datos siempre pertenece a varios clusters, y que se calcula un peso de probabilidad para cada combinación de punto de datos y cluster. Las observaciones pertenecen a múltiples clusters, excepto que las probabilidades crecen o disminuyen dependiendo del peso que tengan asociado.

En el método de agrupación en clusters EM, el algoritmo refina de forma iterativa un modelo de clusters inicial para ajustar los datos y determina la probabilidad de que un punto de datos exista en un cluster. El algoritmo finaliza el proceso cuando el modelo probabilístico ajusta los datos. La función utilizada para determinar el ajuste es el algoritmo de la probabilidad de los datos dado el modelo.

Si durante el proceso se generan clústeres vacíos, o si la pertenencia de uno o varios de los clústeres cae por debajo del umbral especificado, los clústeres con poblaciones bajas se reinicializan en los nuevos puntos y vuelve a ejecutarse el algoritmo EM.

Los resultados del método de agrupación en clusters EM son probabilísticos. Esto significa que cada punto de datos pertenece a todos los clústeres, pero cada asignación de un punto de datos a un clúster tiene una probabilidad diferente. Dado que el método permite que los clústeres se superpongan, la suma de los elementos de todos los clusters puede superar la totalidad de los elementos existentes en el conjunto de entrenamiento. En los resultados del modelo de minería de datos, las puntuaciones que indican soporte se ajustan para tener en cuenta este hecho.

El algoritmo EM es el algoritmo predeterminado utilizado en los modelos de agrupación en clústeres de Microsoft debido a las ventajas que brinda.

- Requiere examinar la base de datos como máximo una vez.
- Funciona incluso si la cantidad de memoria (RAM) es limitada.

- Tiene la capacidad de utilizar un cursor de sólo avance.
- Sus resultados superan los obtenidos por los métodos de muestreo

EM es un algoritmo probabilístico basado en la hipótesis subyacente de que las instancias han sido generadas/extraídas a partir de una o varias distribuciones (asumimos un modelo probabilístico de los clusters). El objetivo es identificar los parámetros que describen dichas distribuciones. Desde un punto de vista probabilístico queremos encontrar el cluster más probable al que los datos pertenecen. Además, las instancias pertenecen a los clusters con cierta probabilidad (soft clustering). Modela los datos usando una combinación de distribuciones de probabilidad (habitualmente Gaussianas/normales). Cada cluster se representa usando una distribución. La distribución controla la pertenencia de las instancias a dicho cluster. Se denominan "mixturas finitas" porque hay un número finito de clusters/distribuciones. Las distribuciones se combinan usando los pesos asociados a los clusters. (ZDRAVKO y LAROSE 2007)

Dados los parámetros que describen el clustering, ¿cómo determinar la probabilidad con que una instancia pertenece a un cluster? La probabilidad de que x pertenezca al cluster A es:

$$Pr(A|x) = \frac{Pr(x|A)\ Pr[A]}{Pr[x]} = \frac{f(x:\mu_A,\sigma_A)P_A}{Pr[x]} \quad (7)$$

Con el caso gaussiano:

$$f(x;\ \mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma}}\ e^{\frac{(x-\mu)^2}{2\sigma^2}} \ldots(8)$$

La *verosimilitud* de una instancia x dado el clustering es:

$$Pr[x\ \ las\ distribuciones] = \sum_i\ Pr[x/\ cluster_i]\ Pr[cluster_i] \quad (9)$$

2.4.1 Algoritmo

Procedimiento iterativo para la estimación de los parámetros *(μ_i, σ_i, p_i), i=1,.k*

i) Fijar valores iníciales para los parámetros

ii) (Expectativa) Para cada instancia calcular la probabilidad de pertenecer a un cluster

iii) (Maximización) Estimar los parámetros que caracterizan las distribuciones a partir de las nuevas probabilidades (hacer más probable esa observación).

iv) Si no hay condición de parada volver al paso ii.

Las probabilidades de los clusters se almacenan como pesos asociadas a las instancias. Estimación de parámetros con pesos *w*. Sea w_i la probabilidad de que la instancia x_i pertenece al cluster A, entonces:

La media se obtiene: $$\mu_A = \frac{w_1x_1+w_2x_2+\cdots+w_nx_n}{w_1+w_2+\cdots+w_n} \quad (10)$$

La varianza se obtiene: $$\sigma_A^2 = \frac{w_1(x_1-\mu)^2+w_2(x_2-\mu)^2+\cdots+w_n(x_n-\mu)^2}{w_1+w_2+\cdots+w_n} \quad (11)$$

El algoritmo *EM*, se nutre con unos valores iníciales que convergen, después de una serie de iteraciones, a un máximo local de la función. Las medias iníciales de cada atributo para cada cluster, las varianzas y covarianzas iníciales de los atributos para cada cluster y las probabilidades a priori iníciales de cada cluster no se pueden generar a la ligera ya que si no se calculan de una forma apropiada, la convergencia del algoritmo se ve comprometida y habría que recurrir a técnicas de regularización. (Garre Rubio y Charro Cubero 2005)

3 Caso de Estudio 1:
Análisis Experimental con Información de Aspirantes

Mathematica provee un lenguaje que será utilizado por sus ventajas gráficas y cualidades de lenguaje basado en la programación funcional (programación declarativa basada en la utilización de funciones matemáticas), que facilitan un rápido y eficiente proceso de construcción de prototipos.

Debido a las capacidades de análisis matemático y los algoritmos que provee la herramienta fue seleccionada para los siguientes análisis experimentales.

Los análisis experimentales, que se presentan a continuación, serán el principio de la minería de datos. Con el objetivo de buscar información que pudiera apoyar a la toma de decisiones. Sin embargo, es posible que algunos de los experimentos nos arrojen información que podría parecer poco relevante.

3.1 Análisis Experimental 1

Entrada: EDONAC, CVEESCORI, RESUOFICIA

Considere el caso de estudio consistente en un conjunto de datos que corresponden a un vector de tres atributos obtenidos de la base de datos de los alumnos que han presentado examen de admisión para ingresar a la Facultad de Informática, correspondiente a los periodos 2003-2008.

Para este análisis experimental se comienza con obtener los datos de entrada que estarán definidos en un espacio tridimensional. Las coordenadas estarán dadas por los mismos datos. El vector se presenta a continuación, aclarando que es una parte debido a que es un vector de 5,000 registros.

{{22.,29,177.25},{22.,5,0.},{22.,94,98.},{9.,509,75.5},{11.,511,95.},{11.,511,95.},{22.,94,135.},{22.,94,135.},{22.,46,76.75},{22.,27,92.25},{22.,88,129.},{22.,88,129.},…,{22.,25,117.25},{11.,411,83.},{11.,411,83.},{22.,25,0.},{22.,13,0.},{24.,27,0.},{22.,68,78.75},{9.,30,0.},{22.,85,0.},{22.,75,116.},{22.,75,116.}}

Una solución al problema usando funciones avanzadas de *Mathematica* consiste en la identificación de clusters como lo ilustra la Figura 3:1. La información que en este caso se está utilizando es el total de alumnos que han hecho solicitud los últimos seis años y han presentado examen para ingresar a la facultad, se pude ver que la población es de diversas partes del país, de diferentes escuelas y las calificaciones son desde cero hasta 150 puntos.

Utilizando un filtro para analizar únicamente la información de los alumnos aceptados en estos años obtenemos los resultados que se muestran en la Figura 3:1. Donde se pueden ver claramente que los grupos se forman con respecto a las calificaciones, las más altas (por arriba de 70 puntos) se localizan con el color purpura y las que está por debajo con azul. Además de que se están concentrando principalmente en el estado 22 que corresponde a Querétaro.

No podemos identificar solo una escuela como la que ha obtenido las mejores puntuaciones, pero las escuelas que han obtenido más de 150 puntos son las que tienen claves 030, 024, 029, es decir Prepa Norte UAQ ,CETIS 16 ,CBTIS 118, respectivamente.

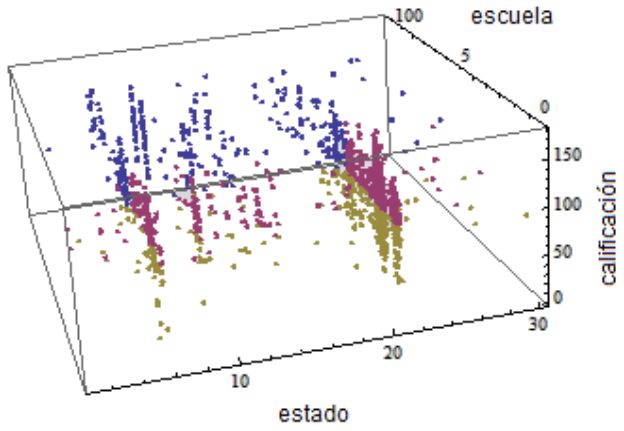


FIGURA 3:1 GRÁFICA CON ESTADO, CALIFICACIÓN Y ESCUELA DE PROCEDENCIA CON K = 3

Sin embargo, se podría imponer un número determinado de clusters k para la separación de los datos de acuerdo a los requerimientos del usuario. Las Figuras 3:1 y 3:2 muestra los casos para k=3 y k=4, respectivamente.

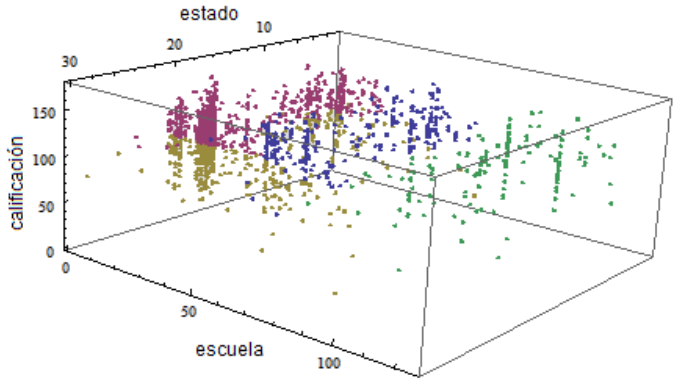


FIGURA 3:2 GRÁFICA CON ESTADO, CALIFICACIÓN Y ESCUELA DE PROCEDENCIA CON K = 4

En la Figura 3:2 encontramos de color amarillo el clúster que identifica las calificaciones más altas. Apreciándose de manera más clara lo que comentábamos arriba.

2.4.1.1 Análisis Experimental 1a

Entrada: EDONAC, RESUOFICIA

El experimento se ha realizado en un espacio de dos dimensiones. Quedando el vector segmentado:

```
{{7,135.25},{6,150.75},{21,117.25},{6,144.25},{9,99.5},{6,149.75},{
21,146.25},{21,46.75},{21,102.},{18,141.75},{4,137.75},{21,108.5},{6,105.
75},{6,141.},{21,117.},{21,115.75},{21,111.75},{6,118.25},{6,117.75},{21,
112.25},{6,22.25},{21,75.75},{21,137.5},{6,108.},{21,158.25},{17,148.},{2
1,105.75},{21,95.5},{6,153.25},{6,141.5},{25,51.5},{6,107.},{21,108.75},{
21,90.5},{21,74.25},{21,116.25},{9,105.},{21,100.5},{21,101.5},{20,139.},
{21,114.25},{6,85.},{9,106.75},{6,108.5},{21,108.5},{21,101.},{25,91.75},
{21,106.},{21,132.25},{6,74.25},{21,122.5},{21,71.25},{21,107.5},{6,122.2
5},{21,122.25},{21,118.25},{21,128.25},{21,99.},{21,128.25},{21,120.25},{
6,118.75},{6,119.25},{21,95.5},{3,102.},{21,47.75},{13,127.25},{21,129.75
},{19,146.75},{21,90.},{9,124.75},{11,99.75},{21,111.25},{13,106.},{15,11
8.25},{6,123.5},{21,138.75},{21,102.},{21,96.75},{21,96.5},{21,100.25},{6
,108.75},{6,133.75},{6,127.75},{6,122.5},{21,97.75},{21,56.25},{14,127.5}
```

Otra vez se presenta una gráfica (Figura 3:3) sin filtrar a los alumnos aceptados, son todos los resultados de los aspirantes a entrar.

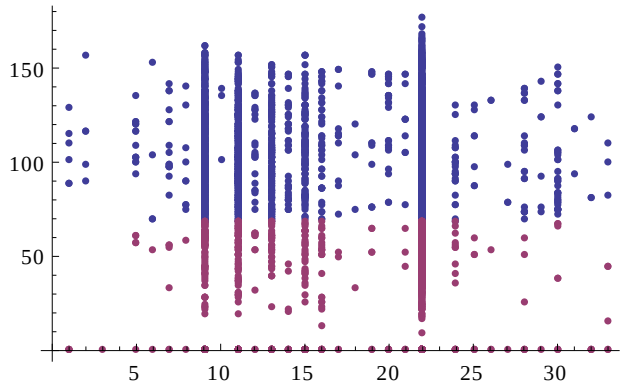


FIGURA 3:3 GRÁFICA EN DOS DIMENSIONES -TODOS LOS ASPIRANTES A INGRESAR CLASIFICADOS EN K=2

Y en la Figura 3:4 solamente los aceptados. Ambas tienen k =2.

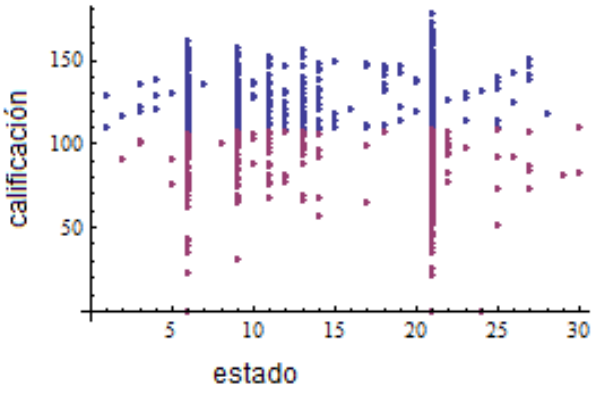


FIGURA 3:4 GRÁFICA ASPIRANTES ACEPTADOS POR ESTADO Y CALIFICACIÓN

En la Figura 3:5 el estado 22 que corresponde a Querétaro es el que tiene mayor población se observa la variedad de puntuaciones obtenidas**.**

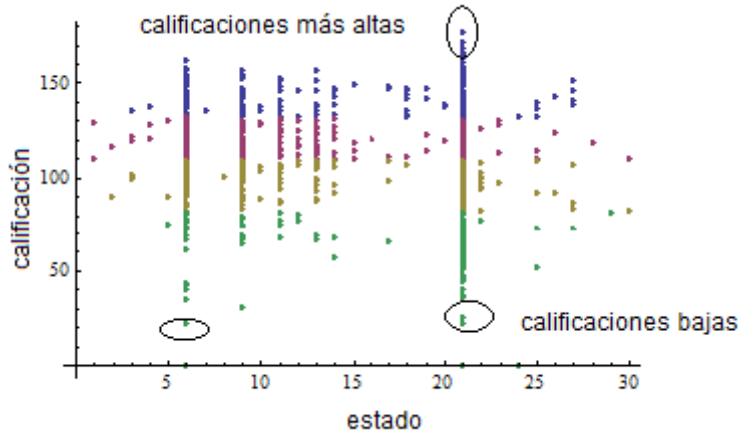


FIGURA 3:5 CALIFICACIONES MÁS BAJAS Y MÁS ALTAS K=4

Se puede concluir que, el estado 6 que corresponde a Colima, y el 22 a Querétaro son los que predominan en puntuaciones altas y bajas. El 9 que es el DF también tiene aspirantes con puntuaciones altas y un dato por debajo de los 50 puntos. Guanajuato que corresponde al 11 tiene la mayoría de sus aspirantes con puntuaciones por arriba de 100. Se han mencionado hasta ahora aquellos estados que tienen una cantidad mayor a diez aspirantes y se observa una dominancia de estados circunvecinos a Querétaro. Sin embargo, es interesante mencionar que los aspirantes que vienen de estados lejanos a nuestro estado, como es el caso del estado 2 Baja California Sur y el 1 Aguascalientes, de los cuales se han tenido pocos aspirantes (menos de 10 en los periodos analizados) han sido aspirantes con puntuaciones mayores a 80 puntos.

2.4.1.2 Análisis Experimental 1b

Entrada: CVEESCORI, RESUOFICIA

El siguiente experimento se ha realizado en un espacio de dos dimensiones, las cuales son clave de escuela de procedencia y puntuación obtenida en el examen de admisión. Quedando una parte del vector de información de la siguiente manera.

{{85,135.25},{84,150.75},{85,117.25},{24,144.25},{24,99.5},{85,149.75},{24,146.25},{85,46.75},{85,102.},{23,141.75},{25,137.75},{17,108.5},{20,105.75},{96,141.},{85,117.},{85,115.75},{85,111.75},{85,118.25},{113,117.75},{39,112.25},{20,22.25},{85,75.75},{23,137.5},{16,108.},{33,158.25},{84,148.},{17,105.75},{24,95.5},{93,153.25},{25,141.5},{103,51.5},{110,107.},{24,108.75},{50,90.5},{25,74.25},{85,116.25},{93,105.},{23,100.5},{37,101.5},{23,139.},{85,114.25},{24,85.},{92,106.75},{85,108.5},{20,108.5},{98,101.},{71,91.75},{33,106.},{24,132.25},{123,74.25},{93,122.5},{47,71.25},{17,107.5},{85,122.25},{22,122.25},{22,118.25},{24,128.25},{25,99.},{22,128.25},{20,120.25},{96,118.75},{85,119.25},{20,95.5}

La gráfica sin filtrar a los alumnos aceptados es la 3:6 y la gráfica 3:7 representa el resultado con filtro de alumnos aceptados, además de normalizadas las escuelas de procedencia.

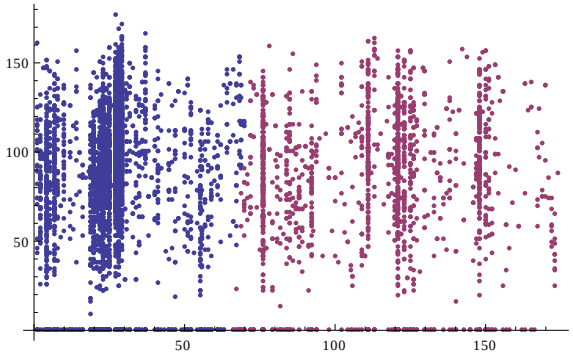


FIGURA 3:6 GRÁFICA TODOS LOS A ASPIRANTES

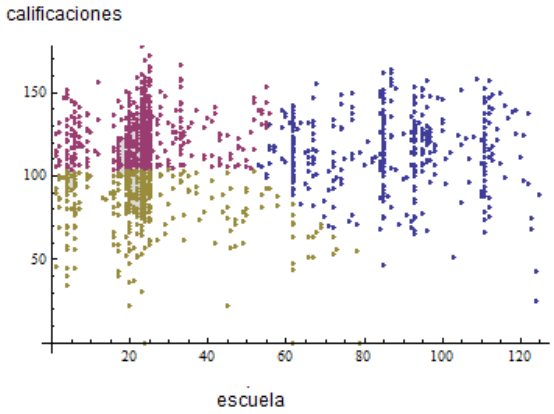


FIGURA 3:7 GRÁFICA ASPIRANTES ACEPTADOS POR ESCUELA Y CALIFICACIÓN, K=3

Se notan homogéneos los resultados es decir, no hay un cluster solo de puntuaciones mayores a 100. También podemos ver que la mayoría de los aspirantes con calificaciones altas no recaen solo en una escuela, pero si la mayoría están en un grupo que van desde la 20 a la 30. Siendo estas Prepa Sur UAQ, Prepa Norte UAQ, CBTIS 116 y algunas escuelas particulares como el Salesiano.

3.2 Análisis Experimental 2

Entrada: RESUOFICIA, CVECARR, FECNA

En este análisis se utiliza información de la edad del aspirante, calificación obtenida y clave del plan al que desea ingresar. Aquí se presenta la versión reducida de la lista.

Short[lista]

{{177.25,18,22},{172.,18,22},{168.75,18,19},{166.5,18,22},{165.,18
,19},{164.25,18,19},{164.,19,19},{163.,18,22},{161.5,18,19},{160.5,19,22}
,{159.,23,22},{158.5,19,22},{158.25,18,22},{158.25,27,22},{156.75,18,22},
{156.5,20,47},{156.5,22,22},{156.,18,22},{156.,18,22},{155.75,18,22},{154
.75,19,19},{154.5,18,22},{154.25,18,19},{153.75,19,47},{153.5,19,22},{153
.25,18,22},{153.25,18,19},{152.75,20,19},{152.5,18,19},{152.5,22,19},{152
.,18,19},{151.5,19,22},{151.25,18,22},{150.75,20,22},{150.75,18,47},{150.
75,18,22},{150.5,23,22},{150.25,18,22},{150.,19,22},{149.75,18,22},{149.2
5,18,22},□927□,{62.25,18,19},{61.25,19,19},{61.25,20,22},{60.25,18,19},{5
9.75,23,19},{59.5,18,30},{59.25,19,19},{58.75,19,30},{58.75,19,30},{58.25
,20,30},{58.,19,19},{57.25,22,22},{57.,22,19},{56.5,22,22},{56.25,23,30},
{56.,18,19},{56.,21,19},{55.75,18,22},{55.25,20,22},{55.,22,22},{54.5,21,
30},{53.,21,19},{52.75,22,22},{52.5,18,48},{52.5,19,19},{51.5,18,19},{51.
,19,48},{50.75,19,19},{47.75,23,30},{47.75,20,48},{46.75,33,19},{45.75,18
,48},{45.,19,19},{44.25,20,19},{44.25,24,19},{44.,18,30},{40.25,18,19},{3
7.5,18,22},{35.75,18,22},{25.25,19,47},{22.,21,19}}

El vector lista inicial corresponde a 1009 registros en triadas. Antes de aplicar la **findClusters** se consideró importante normalizar la información de los planes de carrera, para lo cual se crea por separado el vector nCarr se le asignan los datos correspondientes a los elementos sin repetir, posicionados en la segunda posición de la lista.

$\mathbf{nCarr} = \mathbf{Union}[\mathbf{Map}[\#_{[[\mathbf{2}]]}\&, lista]]$ Obteniendo $\{\mathbf{19}, \mathbf{22}, \mathbf{30}, \mathbf{47}, \mathbf{48}\}$

Se identifica entonces la longitud de este nuevo vector $\mathbf{Range}[\mathbf{Length}[\mathbf{nCarr}]]$ Obteniendo $\{\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}\}$. Y finalmente se hace una correspondencia uno a uno entre ambos vectores:

$\mathbf{Thread}[\{\mathbf{nCarr} = \mathbf{Union}[\mathbf{Map}[\#_{[[\mathbf{2}]]}\&, lista]], Range[Length[nCarr]]\}]$

Obteniendo $\{\{\mathbf{19}, \mathbf{1}\}, \{\mathbf{22}, \mathbf{2}\}, \{\mathbf{30}, \mathbf{3}\}, \{\mathbf{47}, \mathbf{4}\}, \{\mathbf{48}, \mathbf{5}\}\}$

$\mathbf{changes} = \mathbf{Map}[\#_{[[\mathbf{1}]]} \rightarrow \#_{[[\mathbf{2}]]}\&,$

$$\mathbf{Thread}[\{\mathbf{nCarr} = \mathbf{Union}[\mathbf{Map}[\#_{[[2]]}\&, lista]], Range[Length[nCarr]]\}]$$

Lo que se obtendrá como remplazo en la lista es el siguiente mapeo:

$$\{\mathbf{19} \to \mathbf{1}, \mathbf{22} \to \mathbf{2}, \mathbf{30} \to \mathbf{3}, \mathbf{47} \to \mathbf{4}, \mathbf{48} \to \mathbf{5}\}$$

Después de realizar dicho mapeo en la lista inicial. Se aplica la función especializada para encontrar los clusters.

$$\mathbf{c1 = FindClusters}[\mathbf{lista}, \mathbf{3}]$$

La cual arroja un vector c1 ya con los clusters identificados. Graficando se obtiene la Figura 3:8, en la que se puede observar clusters por calificaciones, como en los análisis anteriores los azules son las calificaciones más altas.

$$\mathbf{ListPointPlot3D}[\mathbf{cl}]$$

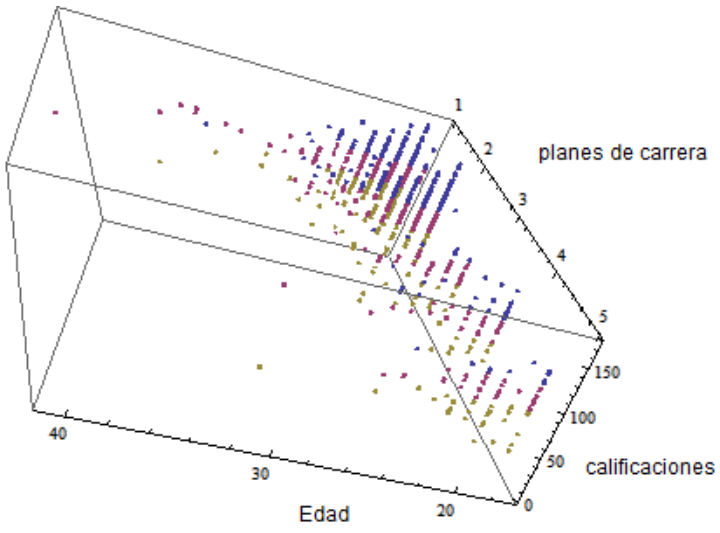


FIGURA 3:8 PLANES DE CARRERA, EDAD Y CALIFICACIÓN. K=3

La relación con la edad es significativa, desde que nos está demostrando que entre más jóvenes los aspirantes obtienen puntuaciones más altas, haciendo énfasis en que no solo son altas, sino que también obtienen calificaciones regulares (rosas) y bajas (amarillas). También, es interesante hacer notar que el rango de edades de los alumnos aspirantes va desde 16 años hasta cerca de los 40 años.

En la figura 3:9 se presentan los cluster claramente por calificaciones. Las calificaciones correspondientes al cluster amarillo están en un rango de (50, 99). El cluster rosa va desde (100, 120) y el azul de (120, 170).

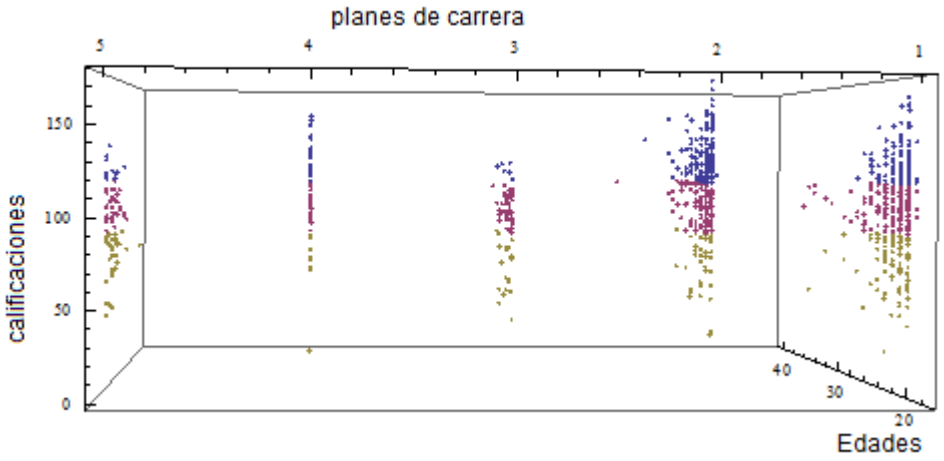


FIGURA 3:9 GRÁFICA PARA IDENTIFICAR CLUSTERS POR CALIFICACIÓN

Con la herramienta se analizan los porcentajes.

$$\mathbf{Map[Length, cl]100/Length[lista]//}N$$

Obteniendo `{34.5887,45.7879,19.6234}`

Los porcentajes que se tienen en las gráficas anteriores nos muestran que el 34.58% son calificaciones altas, el 45.7879% medias y 19.6234% bajas. Como se muestra en la gráfica.

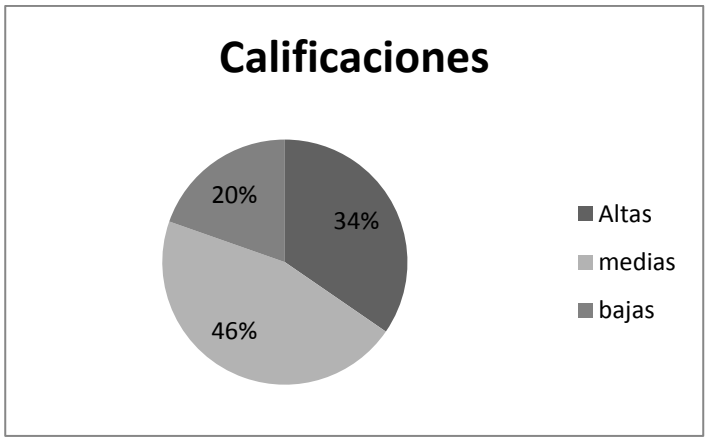


FIGURA 3:10 PORCENTAJES DE CALIFICACIONES

3.2.1.1 Análisis Experimental 2 a

Entrada: EDAD, RESUOFICIA

Análisis de edad y calificación obtenida en el examen de admisión.

Utilizando el siguiente vector de información en formato reducido por la cantidad de información que es.

{20.,66.75},{18.,66.25},{20.,66.},{25.,65.75},{19.,65.5},{19.,65.25},{20.,65.},{20.,65.},{18.,64.5},{19.,63.75},{18.,63.75},{18.,63.5},{20.,63.25},{22.,63.},{18.,63.},{18.,63.},{18.,62.25},{19.,61.25},{20.,61.25},{18.,60.25},{23.,59.75},{18.,59.5},{19.,59.25},{19.,58.75},{19.,58.75},{20.,58.25},{19.,58.},{22.,57.25},{22.,57.},{22.,56.5},{23.,56.25},{18.,56.},{21.,56.},{18.,55.75},{20.,55.25},{22.,55.},{21.,54.5},{21.,53.},{22.,52.75},{18.,52.5},{19.,52.5},{18.,51.5},{19.,51.},{19.,50.75},{23.,47.75},{20.,47.75},{33.,46.75},{18.,45.75},{19.,45.},{20.,44.25},{24.,44.25},{18.,44.},{18.,40.25},{18.,37.5},{18.,35.75},{19.,25.25}}

Los clusters que se encuentran se presentan en la siguiente Figura.

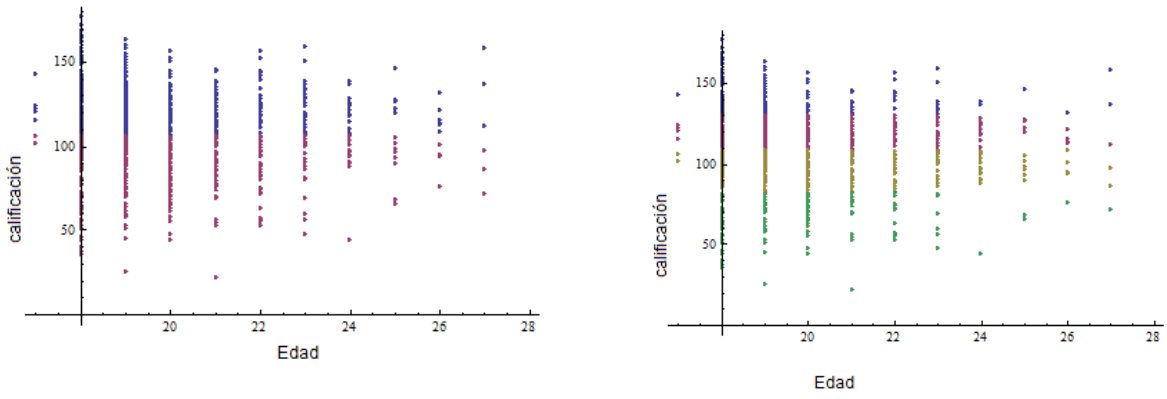


FIGURA 3:11 GRAFICA CON K=2Y K=4, EDAD Y CALIFICACIÓN

Se seleccionaron k=2 y k=4 para tener dos vistas diferentes de clusters.

La Figura 3:11 en k = 2, nos muestra las calificaciones altas en color azul, podemos observar que los alumnos más jóvenes tienen calificaciones más altas, de hecho se encuentran en los alumnos aspirantes de 18 años las puntuaciones más altas. Y la puntuación más baja en los alumnos de 21 y 19 años.

3.2.1.2 Análisis Experimental 2 b

Entrada: RESUOFICIA, CVECARR

Este análisis nos presenta la relación entre calificación obtenida y plan de carrera seleccionado.

Una muestra del vector que se analizará es la siguiente:

{22.,130.5},{19.,130.5},{21.,130.5},{20.,130.25},{18.,130.25},{18.,130.25},{18.,130.},{19.,130.},{19.,130.},{18.,130.},{22.,130.},{22.,130.},{22.,130.},{23.,129.75},{20.,129.5},{18.,129.5},{18.,129.5},{18.,129.5},{19.,129.25},{18.,129.25},{20.,129.},{22.,129.},{20.,129.},{19.,128.75},{19.,128.75},{20.,128.75},{19.,128.75},{18.,128.75},{20.,128.75},{19.,128.75},{18.,128.75},{21.,128.75},{19.,128.5},{18.,128.25}

Los clusters que se obtienen utilizando la función de localización de clusters, es la que se presenta en la Figura 3:12. Donde se puede observar que los planes que tienen mayor demanda son el 1 y el 2, correspondientes a Informática e Ingeniería en Computación, respectivamente. Siendo éste último en el que los aspirantes con mayor calificación solicitan.

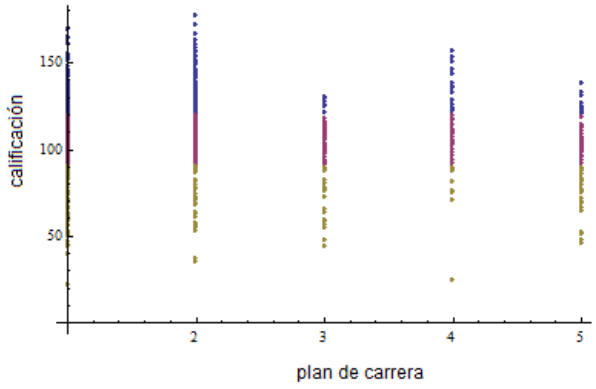


FIGURA 3:12 GRÁFICA CON K=3 PLAN DE CARRERA Y CALIFICACIÓN

3.3 Análisis Experimental 3

Entrada: FECNA, CVEESCORI, RESUOFICIA

El presente análisis está utilizando los atributos de edad El vector inicial para este análisis en formato corto es:

```
{{18.,177.25,29},{18.,172.,31},{18.,168.75,30},{18.,166.5,41},{18.,165.,3
1},{18.,164.25,31},{19.,164.,348},{18.,163.,31},{18.,161.5,348},{19.,160.
5,29},{23.,159.,27},{19.,158.5,30},{18.,158.25,31},{27.,158.25,41},{18.,1
56.75,41},{20.,156.5,15},{22.,156.5,31},{18.,156.,29},{18.,156.,30},{18.,
155.75,31},{19.,154.75,106},{18.,154.5,30},{18.,154.25,31},{19.,153.75,31
},{19.,153.5,41},{18.,153.25,41},{18.,153.25,84},{20.,152.75,349},{23.,56
.25,25},{18.,56.,4},{21.,56.,7},{18.,55.75,114},{20.,55.25,131},{22.,55.,
22},{21.,54.5,25},{21.,53.,112},{22.,52.75,29},{18.,52.5,24},{19.,52.5,27
}}
```

Generando los clusters obtenemos la gráfica que se muestra en la Figura 3:13, donde se presentan dos posiciones diferentes para k=3.

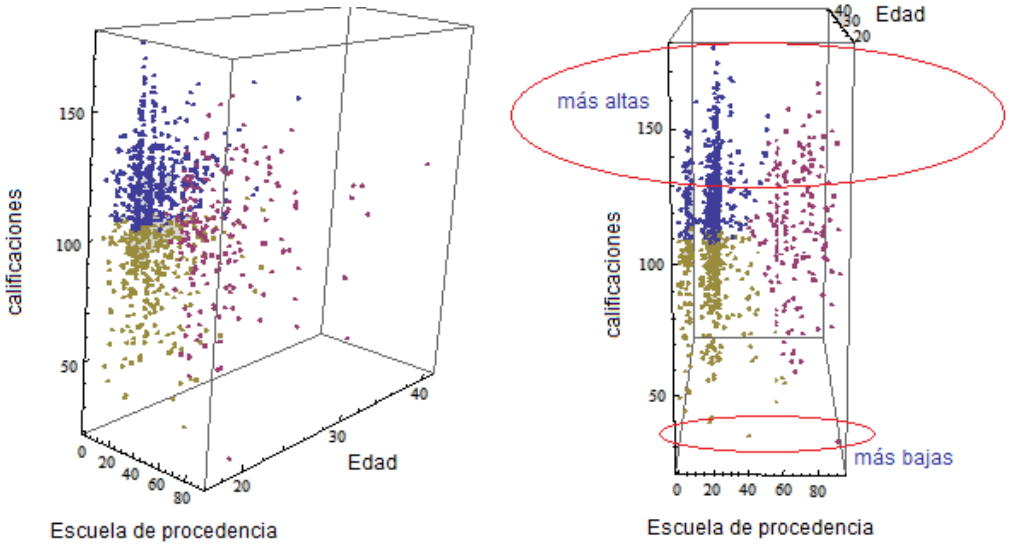


FIGURA 3:13 GRÁFICA CON K = 3 PARA ESCUELA DE PROCEDENCIA, EDAD Y CALIFICACIÓN

En la Figura 3:13a se puede observar que la mayor cantidad de alumnos e ubican entre 18 y 23 años. Se puede identificar también en el cluster azul las calificaciones más altas, aunque comparte esto con algunos puntos del cluster rosa.

En la Figura 3:13b, se marcan las calificaciones más bajas (menores a 50 puntos) proceden de escuelas diferentes, las cuales se presentan en la Tabla 3:1.

TABLA 3-1 ESCUELAS DE PROCEDENCIA CON CALIFICACIONES MÁS BAJAS

Escuelas de procedencia de aspirantes aceptados
CBTIS 118
COL BACH PIE DE LA CUESTA(13)
COL BACH SATELITE
COL BACH STA. ROSA JAUREGUI
COL BACH VILLA CORREGIDORA
COL BACHILLERES (17)
CONALEP QUERETARO
CONALEP QUERETARO
PART HENRY FORD
PART ISCCA
PREPA NORTE UAQ.

Y las puntuaciones más altas provienen de una gama amplia de escuelas. Por lo que no se puede concluir que los mejores alumnos vengan solo de una escuela.

El análisis en dos dimensiones que corresponde a edad y escuela se presenta en el análisis experimental 3a.

El análisis en dos dimensiones correspondiente a calificaciones y edad se encuentra en el experimento 2.

3.3.1.1 Análisis Experimental 3 a

Entrada: FECNA, CVEESCORI

Este análisis identifica las escuelas de procedencia y las edades de los alumnos.

El vector en su versión corta queda de la siguiente manera:

```
{{18.,21},{18.,23},{18.,22},{18.,30},{18.,23},{18.,23},{19.,78},{18.,23},
{18.,78},{19.,21},{23.,20},{19.,22},{18.,23},{27.,30},{18.,30},{20.,11},{
22.,23},{18.,21},{18.,22},{18.,23},{19.,62},{18.,22},{18.,23},{19.,23},{1
9.,30},{18.,30},{18.,50},{20.,79},{18.,20},{22.,21},{18.,22},{19.,22},{18
.,21},{20.,21},{18.,22},{18.,22},{23.,17},{18.,22},{19.,22},{18.,70},{18.
,21},{19.,22},{19.,61},{33.,22},{25.,76},{21.,21},{20.,5},{22.,21},{19.,2
3},{19.,23},{18.,23},{21.,87},{21.,15},{18.,18},{18.,23},{19.,76},{19.,23
},{22.,50},{20.,6},{17.,19},{20.,82},{18.,21},{22.,6},{18.,68},{19.,89},{
19.,16},{20.,20},{18.,23},{20.,25},{18.,56},{18.,70},{18.,22},{18.,7},{18
.,23},{18.,21},{18.,23},{22.,29},{21.,23},{19.,23},{21.,23},{18.,30},{23.
,56},{19.,34},{19.,48},{19.,49},{24.,76},{18.,22},{18.,18},{21.,6},{21.,8
2},{19.,70},{27.,21}
```

La gráfica presenta cuatro clusters y permite visualizar de manera más precisa que la mayoría de los alumnos tienen entre 18 y 22 años

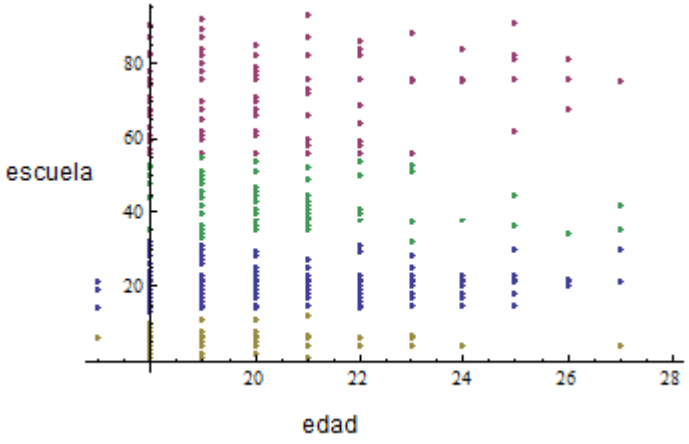


FIGURA 3:14 EDAD Y ESCUELAS DE PROCEDENCIA, CON K=4

3.3.2 Análisis Experimental 4

Entrada: CVECARR, SEXO, RESUOFICIA

El presente análisis experimental podría generar controversias, debido a que sería posible hacer el planteamiento de cuestiones cómo: ¿en que podría beneficiar a la institución el conocer por género las calificaciones obtenidas en el examen de admisión? O bien, sería realmente interesante diferenciar cuál de los dos géneros seleccionan cada plan de estudio.

Tracy Camp en su famoso artículo para la Nacional Science Foundation (Camp 1997) señala cómo el porcentaje de mujeres en las carreras de informática en Estados Unidos había ido creciendo de 1975 a 1985 (alcanzando su máximo en ese año con un 35 % de mujeres), mientras que en la segunda mitad de los ochenta empezó a descender hasta el 32 % en 1988. Las cosas han empeorado desde entonces, y en 1999 encontramos no más de un 28 % de mujeres que acaban su licenciatura en informática en Estados Unidos (ni que decir tiene que el porcentaje de mujeres que acababan un doctorado en informática era aún menor: sólo un 12 %).

Veamos en nuestro caso de análisis, como se comportan los datos.

El vector a analizar queda de la siguiente manera:

{{22,0,177.25},{22,1,172.},{19,1,168.75},{22,1,166.5},{19,1,165.},{19,1,164.25},{19,1,164.},{22,1,163.},{19,1,161.5},{22,1,160.5},{22,1,159.},{22,1,158.5},{22,1,158.25},{22,1,158.25},{22,1,156.75},{47,1,156.5},{22,1,156.5},{22,1,156.},{22,0,156.},{22,1,155.75},{19,0,154.75},{22,1,154.5},{19,0,154.25},{47,1,153.75},{22,1,153.5},{22,1,153.25},{19,0,153.25},{19,1,152.75},{19,1,152.5},{19,1,152.5},{19,1,152.}}

Posteriormente, se hace una relación a los planes de carrera, buscando que el vector quede más simple para analizar, por lo que como se hizo en el análisis experimental 2 se normaliza la información

Aplicando la función **findClusters**, queda la siguiente gráfica.

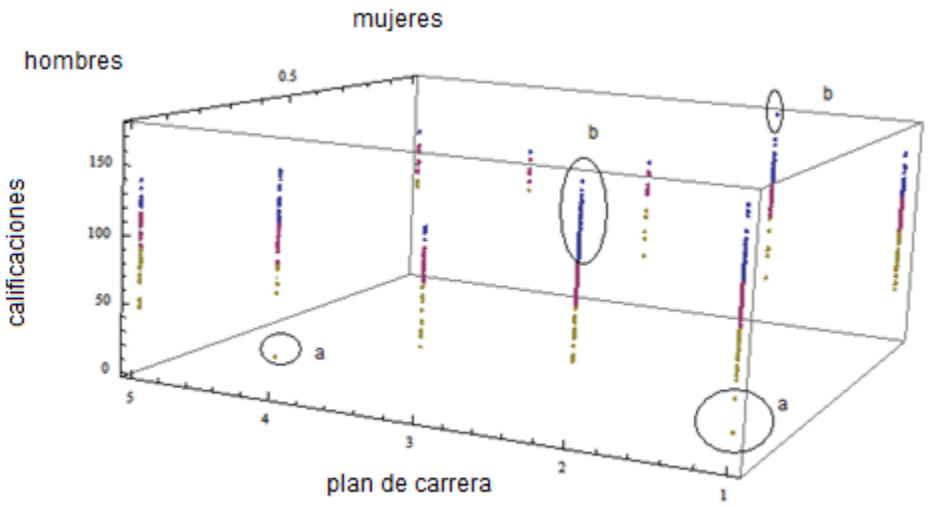


FIGURA 3:15 ANÁLISIS EXPERIMENTAL PLAN DE CARRERA, GÉNERO Y CALIFICACIÓN OBTENIDA EN EL EXAMEN DE ADMISIÓN

Es interesante notar que las calificaciones bajas correponden más al genero masculino en los planes de Informática e Ingeniería de Software.

En la Figura 3:15 podemos observar lo siguiente: los puntos señalados con la letra *a* corresponden a las calificaciones más bajas y se encuentran en el género masculino (1). Y entre el cluster azúl, que corresponde a las calificaciones más altas, señalado con la letra b, aunque se puede ver que tanto género masculino como femenino se encuentran en el cluster, también es claro que entre las calificaciones más altas encontramos al género femenino. Y además, ambos en el plan de Ingeniería en computación.

También la gráfica nos muestra que en plan perteneciente a la carrera de Ingeniería en Telecomunicaciones, hay poca demanda por parte del género femenino.

3.3.2.1 Análisis Experimental 4a

Entrada SEXO, RESUOFICIA

Como se puede observar en la Figura 3:16, confirmamos lo que apreciábamos en el experimento anterior, las calificaciones más altas las obtienen los aspirantes del género femenino, sin embargo son más dispersas. En cambio el género masculino presenta calificaciones más bajas.

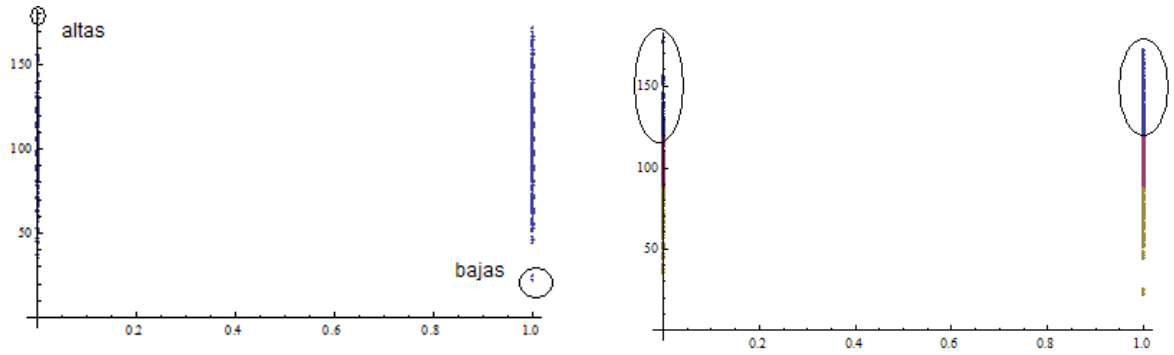


FIGURA 3:16 GRÁFICAS CALIFICACIONES POR GÉNERO (MASCULINO, FEMENINO)

3.4 Análisis Experimental 5

Entrada CVECARR, RESUOFICIA, CVEESCORI

La entrada de este análisis experimental es la clave de la carrera a la que aspira el alumno, la puntuación obtenida en el examen de admisión y la clave de la escuela de la que procede. Así queda un vector con tres atributos, por lo tanto de tres dimensiones.

Quedando una parte del vector de información de la siguiente manera.

{{22,177.25,29},{22,172.,31},{19,168.75,30},{22,166.5,41},{19,165,31},{19,164.25,31},{19,164.,348},{22,163.,31},{19,161.5,348},{22,160.5,29},{22,159.,27},{22,158.5,30},{22,158.25,31},{22,158.25,41},{22,156.75,41},{47,156.5,15},{22,156.5,31},{22,156.,29},{22,156.,30},{22,155.75,31},{19,154.75,106},{22,154.5,30},{19,154.25,31},{47,153.75,31},{22,153.5,41},{22,153.25,41},{19,153.25,84},{19,152.75,349},{19,152.5,27},{19,152.5,29},{19,152.,30},{22,151.5,30},{22,151.25,29},{22,150.75,29},{47,150.75,30},{22,150.75,30},{22,150.5,24},{22,150.25,30},{22,150.,30},{22,149.75,130},{22,149.25,29},{21,58.75,30},{31,58.75,30},{22,58.25,30},{72,58.,19},{4,57.25,22},{70,57.,19},{26,56.5,22},{25,56.25,30},{4,56,19},{7,56.,19},{114,55.75,22},{131,55.25,22},{22,55.,22}}

Y la gráfica asociada es la que se presenta en la Figura 3:17, por defecto arroja el algoritmo cuatro clusters. Se presentan claramente las cinco carreras, es notable que las que corresponden a las claves entre 19 y 22, Informática e Ingeniería en computación respectivamente, son las que tiene mayor población.

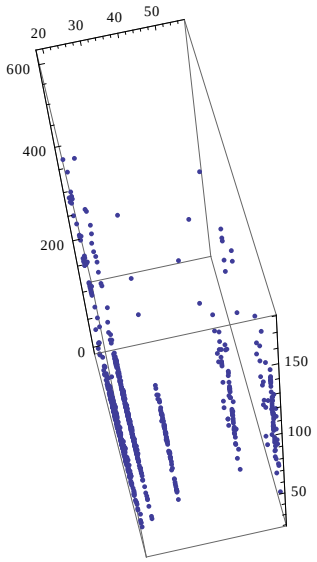


FIGURA 3:17 GRÁFICA ANÁLISIS EXPERIMENTAL 7

Para entender esta gráfica es necesario identificar los ejes. Primero se presenta el que respecta a la clave de carrera, cuyo rango se encuentra entre [19-48] (ver la Tabla 2), en la gráfica es el eje correspondiente a ***y***, donde los valores van desde 0 a 40.

En cuanto al eje ***x***, vemos una amplitud de rango que va desde 0 hasta 600, debido a que las claves de las escuelas tienen ese comportamiento. Sin aplicar aún el algoritmo de clustering podemos observar una conglomeración entre las escuelas con claves de 50 a 150.

Finalmente en el eje correspondiente a ***z***, que da la profundidad del cubo. Se presenta la puntuación obtenida en el examen de admisión. El rango esta variando de 0 a 150 puntos.

A continuación se presenta la gráficas obtenidas con diferentes selección de clusters.

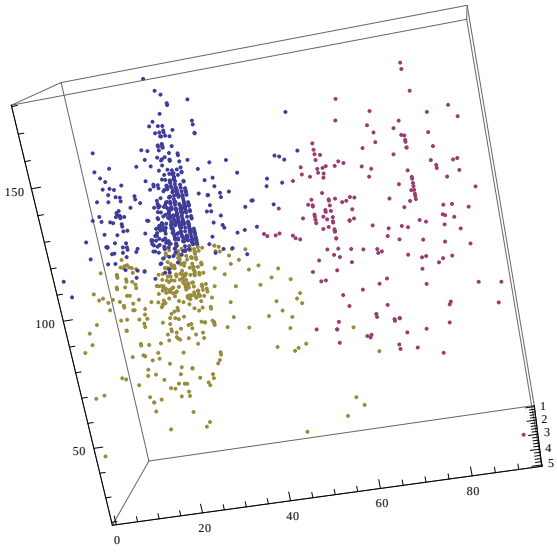


FIGURA 3:18 GRÁFICA DE CLUSTERS CON K = 3

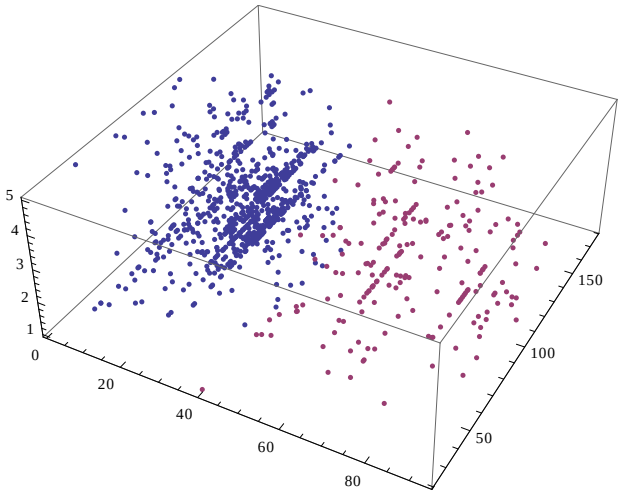


FIGURA 3:19 GRÁFICA DE CLUSTERS CON K = 2

Se puede observar en las figuras 3:18 y 3:19, que en los tres clusters hay resultados muy variables en lo que respecta a la puntuación obtenida. Sin embargo, en el cluster azul se presenta una conglomeración y evidentemente son las calificaciones más altas. Las más bajas son las que se identifican con el color amarillo y las rosas como aquellos casos atípicos. Donde nos indica que la mayoría de los aspirantes seleccionan las carreras de Informática e Ingeniería en Computación. Notándose también que las calificaciones más altas son de alumnos que aspiran a la carrera de ingeniería en computación.

Las escuelas de la 1 a la 40 presentan a los mejores aspirantes por las calificaciones obtenidas en el examen. Sin embargo, también se nota que hay en todas las escuelas alumnos con buenas y malas calificaciones.

3.4.1 Análisis Experimental 4a, dos dimensiones

Entrada CVECARR, CVEESCORI

En este caso se analizan solo dos variables. En estas dos dimensiones queda la carrera que se seleccionó sobre el eje de la **Y**. Así como la escuela de origen, en el eje de las **X**. Cabe comentar que tanto las carreras que han quedado numeradas del 1 al 5, como las escuelas de procedencia también fueron ajustadas, para así lograr más claridad en la gráfica.

La gráfica asociada en la Figura 3:20:

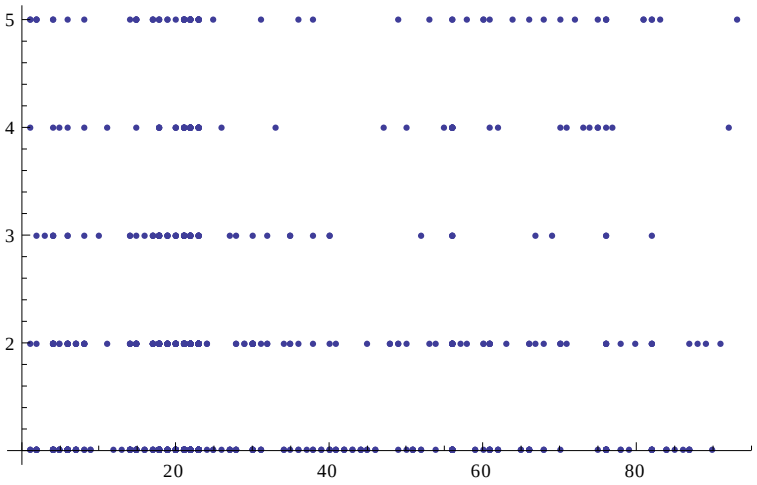


FIGURA 3:20 GRÁFICA EN DOS DIMENSIONES, ESCUELA DE ORIGEN Y CARRERA SELECCIONADA

En la Figura 3:21 se pueden observar los cuatro clusters que se formaron. Verticalmente se organizan por escuela de origen y horizontalmente por planes de carrera.

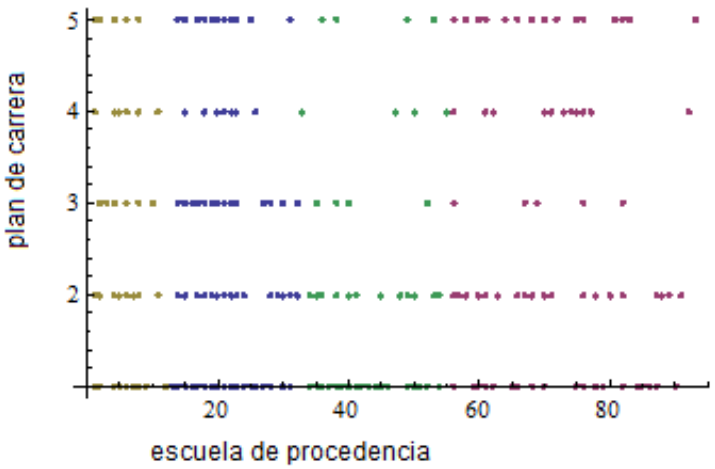


FIGURA 3:21 GRÁFICA CON K= 4, ESCUELA DE PROCEDENCIA Y PLAN DE CARRERA SELECCIONADO POR ASPIRANTES

El rango de las escuelas que son las que más hacen solicitud a nuestra facultad se encuentra entre la 50-90. Correspondiendo a los datos que se presentan en la Tabla 3-1. También podemos identificar que los alumnos de dichas escuelas prefieren el plan de Informática así como la Ingeniería en Computación.

TABLA 3-2 ESCUELAS CON MAYOR CANTIDAD DE ASPIRANTES A LA FACULTAD DE INFORMÁTICA-

Escuela	Número de registros
Colegio de Bachilleres Pie de la cuesta	46
Escuela Abierta, Secretaría de Educación Pública	27
Escuela Pública de Guanajuato	26
Escuela Privada de Guanajuato	5
Colegio de Bachilleres San Joaquín	13

En el rango de 30-50 son las escuelas donde los alumnos aspirantes prefieren la carrera de ingeniería de software. En la siguiente Tabla se muestran los nombres de las escuelas.

TABLA 3-3 ESCUELAS CON MÁS SOLICITUDES PARA LA CARRERA DE INGENIERÍA DE SOFTWARE.

Escuela	Número de registros
Colegio de Bachilleres Satélite	88
Prepa Norte	12
Colegio de Bachilleres Pie de la cuesta	8

3.4.2 Análisis Experimental 4b, dos dimensiones

Entrada CVECARR, RESUOFICIA

En este análisis experimental estaremos verificando en qué planes de estudio se encuentran las mejores puntuaciones obtenidas en el examen de admisión.

El vector en su versión corta para este análisis se presenta a continuación:

```
    {{2,177.25},{2,172.},{1,168.75},{2,166.5},{1,165.},{1,164.25},{1,164
.},{2,163.},{1,161.5},{2,160.5},{2,159.},{2,158.5},{2,158.25},{2,158.25},
{2,156.75},{4,156.5},{2,156.5},{2,156.},{2,156.},{2,155.75},{1,154.75},{2
,154.5},{1,154.25},{4,15}3.75},{2,153.5},{2,153.25},{1,153.25},{1,152.75}
,{1,152.5},{1,152.5},{1,152.},{2,151.5},{2,151.25},{2,150.75},{4,150.75},
{2,150.75},{2,150.5},{2,150.25},{2,150.},{2,149.75},{2,149.25},{1,148.},{
1,146.5},{2,146.25},{4,146.25},{1,145.5},{2,145.25},{2,145.25},{2,145.},{
1,144.75},{2,144.75},{1,144.75},{2,144.5},{1,144.5},{2,144.5},{2,144.25},
{1,144.},{4,144.},{1,143.5},{1,143.5},{1,143.5},{2,143.25},{2,142.5},{1,1
42.5},{2,142.},{1,141.5},{2,141.5},{1,141.5},{1,141.5},{2,141.5},{2,141.5
},{2,141.25},{2,140.75},{2,140.75},{2,139.5},{2,139.5},{2,139.5},{1,139.}
,{2,139.},{2,139.},{1,139.},{2,139.},{1,138.75},{2,138.75}
```

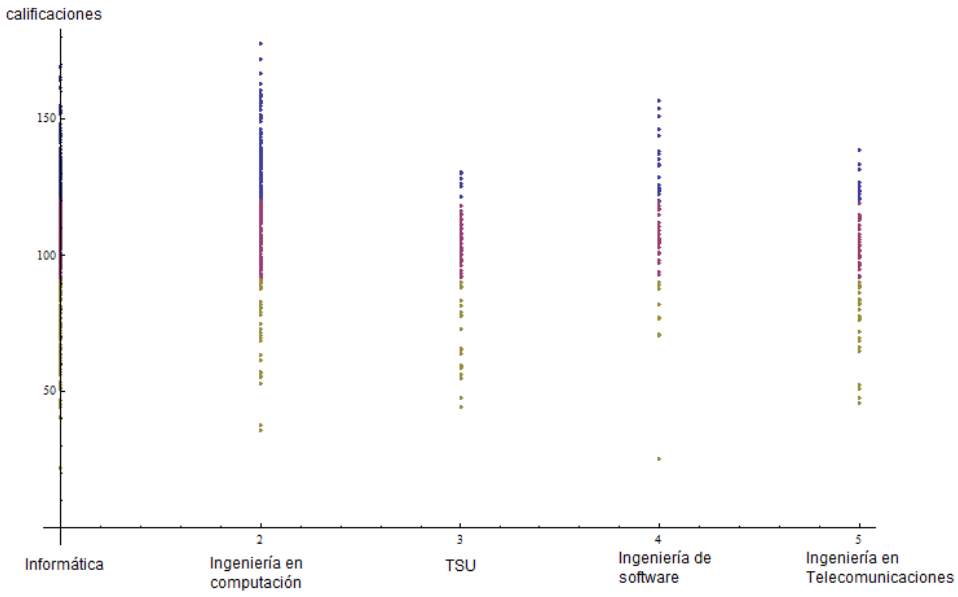


FIGURA 3:22 GRÁFICA QUE PRESENTA PLANES DE ESTUDIO Y RESULTADO OFICIAL DE ALUMNOS ACEPTADOS

La Figura 3:22 presenta las calificaciones más altas que se encuentran en el plan de estudio perteneciente a la carrera de Informática e Ingeniería en Computación. El plan de estudios de Ingeniería de Software presenta el mismo fenómeno, aún cuando es un plan de reciente creación, razón por la cual se visualiza menos continuidad.

Se puede observar que los aspirantes al plan de estudios de Técnico Superior Universitario son alumnos con puntuación media a baja.

3.4.3 Análisis Experimental 6

Entrada FOLIO, RESULCAL, ACEPTADO

Para este experimento se estará utilizando 5869 registros. Por lo que el vector que representa los datos en su versión corta es:

{{17013,177.25,1},{17019,0.,0},{17020,98.,0},{17021,75.5,0},{17022,95.,0},{17023,95.,0},{17024,135.,1},{17025,135.,0},{17026,76.75,0},{17027,92.25,0},{17028,129.,1},{17029,129.,0},{17031,87.25,0},{17032,87.25,0},{17034,130.,0},{17035,137.25,1},{17036,137.25,0},{17037,45.75,0},{19754,106.25,1},{19755,106.25,0},{19756,108.5,1},{19760,67.,0},{19761,55.25,0},{19762,55.25,0},{19766,117.25,1},{19767,117.25,0},{19771,83.,0},{19772,83.,0},{19778,0.,0},{19806,0.,0},{19843,0.,0},{19851,78.75,0},{19874,0.,1},{19919,0.,0},{19973,116.,0},{19974,116.,1}}

Es interesante comentar que en este último experimento se están utilizando los números de folio de los alumnos que presentaron el examen de admisión, aún cuando no hayan sido aceptados por la calificación obtenida.

Fue necesario normalizar la parte de aceptado, sustituyendo si = 1 y no = 0.

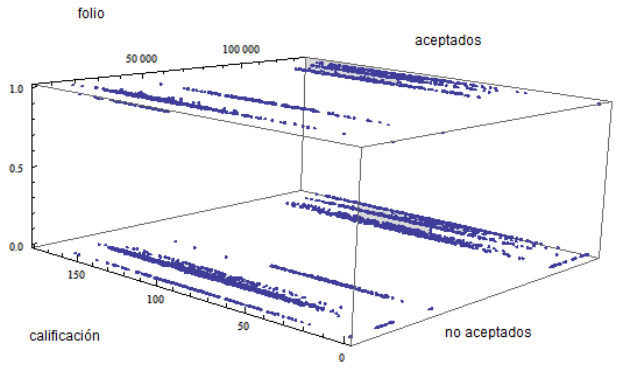


FIGURA 3:23 GRÁFICA DE CALIFICACIÓN, ACEPTADOS Y NO ACEPTADOS

En la Figura 3:23 se puede observar un agrupamiento natural, por aceptados y no aceptados. Los porcentajes de cada cluster se muestran en la Figura 3:24.

FIGURA 3:24 PORCENTAJE DE ALUMNOS ASPIRANTES ACEPTADOS Y NO ACEPTADOS

Aplicando la función de localización de clusters, se obtienen dos grupos que no corresponden al que se mencionaba. Más bien corresponden a folios.

Se presentan otras dos gráficas donde la cantidad de clusters k, es 3 y 4 en la Figura 3:26.

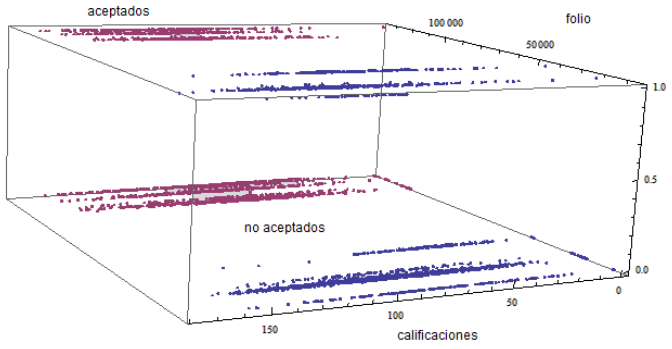


FIGURA 3:25 GRÁFICA DE FOLIOS, CALIFICACIONES Y ACEPTADOS Y NO ACEPTADOS CON K = 2

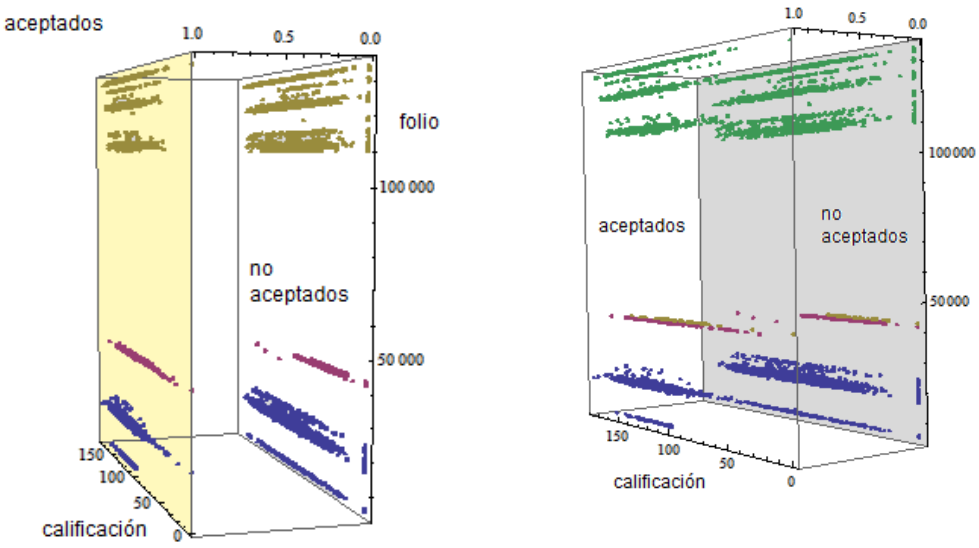


Figura 3:26 Gráficas con rotación de aspirantes aceptados y no aceptados
k = 3 y k = 4

3.5 Resultados y Discusión

Es interesante la manera en la que la herramienta Mathematica permite minar la información. La flexibilidad que nos provee, permite utilizar una sencilla base de datos que no es necesario que sea relacional. Y aún con estas características que un minador experimentado podría comentar como demasiado simples, permite hacer un análisis profundo de la información.

Consideramos que la capacidad del algoritmo matemático implementado y la capacidad gráfica son los puntos fuertes de la herramienta. Es importante mencionar que ésta herramienta basa su método **FindClusters** en algoritmos que aplican distancias Euclidianas.

Una posible restricción para utilizarla es la complejidad que requiere un lenguaje de alto nivel, como el que se maneja Mathematica.

En cuanto a los resultados del análisis podríamos comentar que aún cuando la base de datos con la que se trabajó es relativamente pequeña y no tiene una estructura relacional, nos permitió encontrar datos interesantes como las escuelas que a lo largo de estos años han traído a la Facultad de Informática la mayor cantidad de aspirantes con resultados altos en su examen de admisión.

También nos permitió visualizar los planes de carrera que se caracterizan por ser solicitados por aspirantes cuya característica es puntuación alta en el resultado oficial, que en este caso fueron los planes de estudio de Informática e Ingeniería en Computación. Es interesante notar que a pesar de que tiene poco tiempo el plan de carrera de Ingeniería de Software, 2007, también es solicitado por alumnos con puntuación arriba de 90 puntos.

Por género se identificó que a pesar de que hay pocas alumnas en el plan de estudios de Ingeniería en Computación, ellas sobre salen con puntuaciones altas.

Considero interesante que como análisis posterior, se analizará el desempeño académico de los alumnos cuyas puntuaciones en el examen de admisión fueron altas, para revisar si hay continuidad en las calificaciones obtenidas dentro de la Facultad.

4 Análisis experimental Clustering utilizando Microsoft Business Intelligence Development Studio

La empresa Microsoft es importante en el desarrollo de aplicaciones, en software cuya característica es la facilidad para su uso, a lo cual ellos nombran “amigables”. Otra de sus cualidades es la cantidad de documentación que se encuentra disponible, libros electrónicos, tutoriales, que permiten aprender la herramienta rápidamente.

El manejador de bases de datos de la empresa Microsoft, tiene un conjunto de herramientas (conocido como SQL Server Managment Studio), entre ellas se encuentra una dedicada al Análisis de Información. Dicha herramienta provee diferentes algoritmos, entre ellos está el Algoritmo Microsoft Clustering.

Clustering es regularmente utilizado para definir segmentos de mercado. Como se ha comentado en la presente tesis, existen diferentes algoritmos para hacerlo. Una de las más antiguas k-medias al igual que EM son los dos algoritmos que Microsoft implementa en la herramienta de análisis de información.

En el siguiente caso de estudio haremos experimentos similares a los que se realizaron utilizando Mathematica. Para poder hacer comparaciones en los resultados. Como se mencionó anteriormente la herramienta de Microsoft permite dos algoritmos de clustering como base: EM y k-medias. Ambos algoritmos pertenecen a la categoría de Clustering Probabilístico, del cual se ha explicado en la sección de algoritmos. Son algoritmos que por utilizar aproximaciones probabilísticas son populares en las herramientas de software de minería de datos.

La herramienta de Inteligencia de Negocios (Microsoft Business Intelligence Development Studio) permite unas variaciones de estos dos algoritmos.

TABLA 4-1 TIPOS DE ALGORITMOS DE CLUSTERING SOPORTADOS POR SQL 2008

Método
EM escalable
EM no escalable
K-medias escalable
K-medias no escalable

La implementación de Microsoft proporciona dos opciones: EM escalable y no escalable. De forma predeterminada, en EM escalable, los primeros 50.000 registros se utilizan para inicializar el examen inicial. Si esta operación se realiza correctamente, el modelo sólo utiliza estos datos. Si el modelo no se puede ajustar con 50.000 registros, se leen otros 50.000. En EM no escalable, se lee el conjunto de datos completo independientemente de su tamaño. Este método puede crear clústeres más precisos, pero los requisitos de memoria pueden ser significativos. Dado que EM escalable funciona en un búfer local, recorrer los datos en iteración es mucho más rápido, y el algoritmo hace un mejor uso de la caché de memoria de la CPU que EM no escalable. Es más, EM escalable es tres veces más rápido que EM no escalable, incluso si todos los datos caben en la memoria principal. En la mayoría de casos, la mejora en el rendimiento no significa una reducción de la calidad del modelo completo.

La implementación de Microsoft adapta el método k-mediana a atributos discretos de clúster mediante el uso de probabilidades. El algoritmo k-medias proporciona dos métodos para realizar un muestreo en el conjunto de datos: k-medias no escalable, que carga el conjunto de datos completo y realiza una pasada de agrupación en clústeres, y k-medias escalable, donde el algoritmo utiliza los primeros 50.000 casos y lee más casos únicamente si necesita más datos para lograr un buen ajuste del modelo a los datos.

4.1 Preparando la herramienta para el análisis de la información

La manera en la que se prepara la información para utilizar la herramienta se describe brevemente a continuación:

1. Se debe de contar con una base de datos en el manejador SQL Server 2008.

2. Es necesario levantar los servicios de SQL y Analysis Services. Se muestran dichos servicios en la Figura

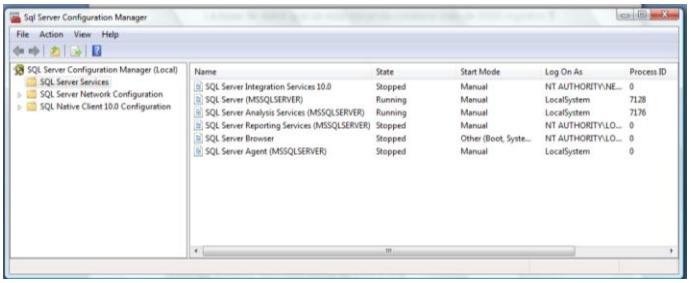


FIGURA 4:1 LEVANTANDO SERVICIOS SQL Y ANALYSIS SERVICES

3. A continuación se inicia la herramienta Business Intelligence Develoment Studio. Si es la primera vez que se utilizará el proyecto, hay que generarlo.

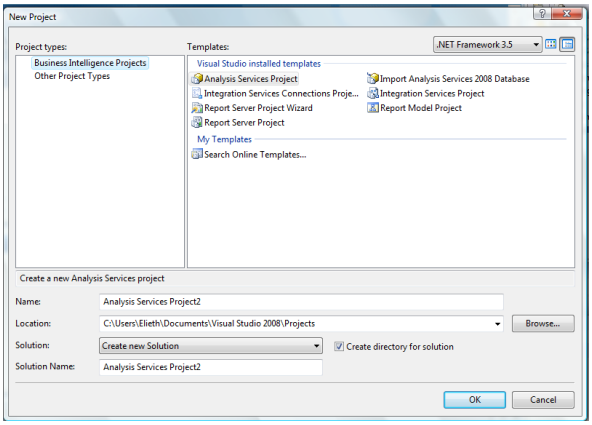


FIGURA 4:2 INICIADO EL PROYECTO DE ANALYSIS SERVICES PROJECT

4. Al generar el proyecto se selecciona la base de datos que deberá estar previamente instalada en SQL.

5. En la siguiente etapa es necesario crear la estructura de minería de datos. Puede ser tan compleja como se quiera, se pueden crear los cubos que se analizarán. En el caso del análisis experimental que se presenta en este proyecto, se está utilizando una tabla como se muestra en la Figura 4:3.

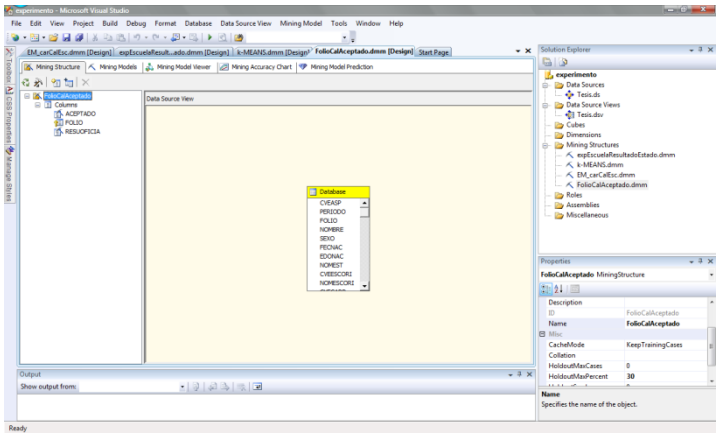


FIGURA 4:3 ESTRUCTURA A MINAR

6. Mientras que una estructura de minería de datos define el dominio de datos, un modelo de minería de datos define el modo de aplicar los datos de ese dominio a un problema determinado. Una vez creada una estructura de minería de datos, puede agregar varios modelos de minería de datos a dicha estructura.

Es necesario seleccionar un algoritmo de minería de datos. El algoritmo de minería de datos es el mecanismo que crea un modelo de minería de datos. Para crear un modelo, un algoritmo analiza primero un conjunto de datos y luego busca patrones y tendencias específicos. El algoritmo utiliza los resultados de este análisis para definir los parámetros del modelo de minería de datos. A continuación, estos parámetros se aplican en todo el conjunto de datos para extraer patrones procesables y estadísticas detalladas. Entre los algoritmos que la herramienta permite seleccionar se encuentran: Clustering, Árboles de Decisión, Regresión Lineal, Redes Neuronales, entre otros. (Figura 4:4)

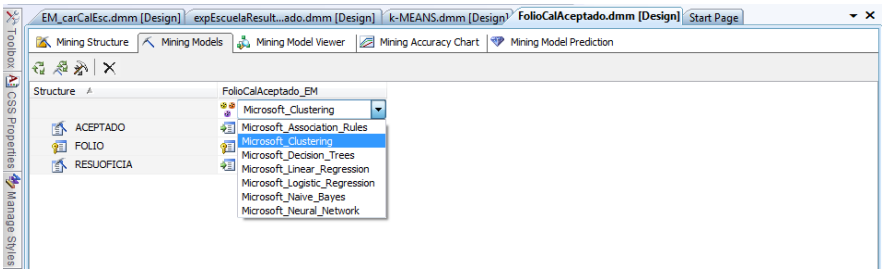


FIGURA 4:4 ALGORITMOS DE MINERÍA DE DATOS.

7. Finalmente queda ejecutar el algoritmo y analizar los resultados. Los cuales se pueden observar en la vista de Modelo Minado (Minning Model Viewer).

2.4.2 Diferentes Vistas de la información.

Después de ejecutar el algoritmo se pueden utilizar las diferentes vistas de los resultados para analizar la información. La herramienta permite ver los resultados de manera gráfica o con una vista de árbol (Figura 4:5) en la cual se observan la descripción del nodo, las probabilidades de pertenencia de cada dato, en el cluster seleccionado, entre otros datos. Para propósitos del proyecto se estará utilizando la gráfica.

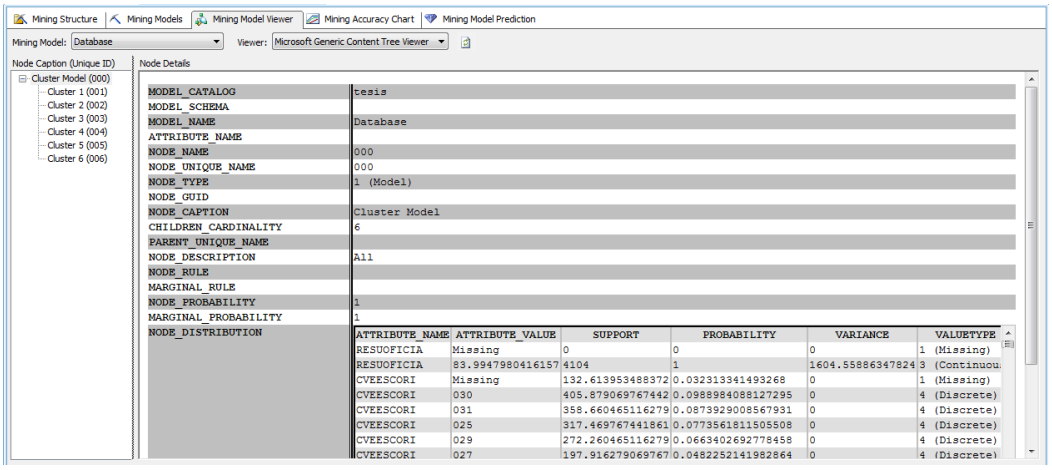
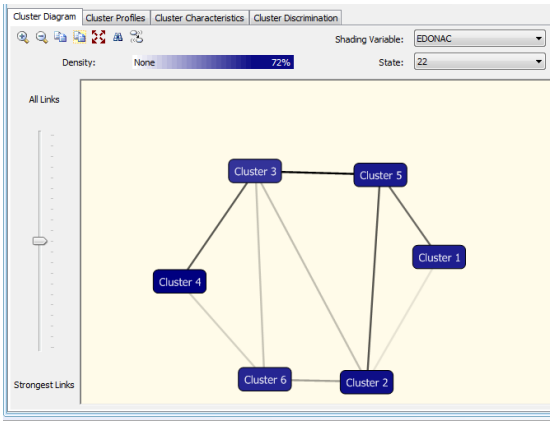


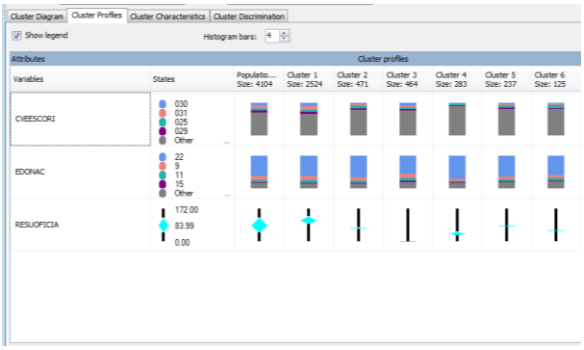
FIGURA 4:5 VISTA EN FORMA DE ÁRBOL

También la herramienta nos presenta los resultados en diferentes visualizadores de clusters, debido a que es el tipo de algoritmos que se seleccionó. Entre las vistas que se presentan están:

1) diagrama de clusters, 2) perfil del cluster (Figura 4:6), 3) características del cluster y 4) complemento del cluster (Figura 4:7).

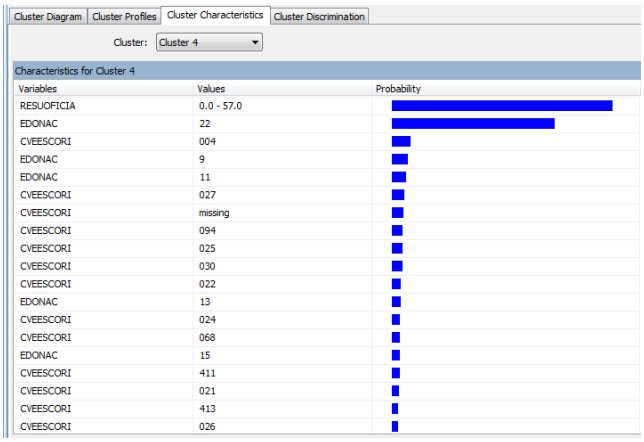


(1) Diagrama

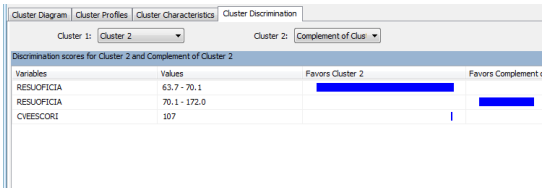


(2) Perfil

FIGURA 4:6 DIAGRAMA DE CLUSTERS Y PERFIL



(3) Características



(4) Complemento

FIGURA 4:7 CARACTERÍSTICAS Y COMPLEMENTO DEL CLUSTER

En los análisis se estarán utilizando el diagrama de clusters y las características del cluster como las vistas que nos dan información suficiente para el análisis.

Al aplicar el algoritmo EM, se generan diez clusters por defecto, sin embargo estos parámetros serán cambiados para lograr un conjunto de clusters, parecido a los logrados en el experimento 1 utilizando Mathematica.

4.2 Caso Experimental utilizando EM

4.2.1 Análisis Experimental 1: resultados obtenidos por los aspirantes, plan de estudio seleccionado y plan de estudios seleccionado

Entrada: aceptado, clave carrera (plan de estudios), folio (identificador único del alumno), resultado oficial (Figura 4:8).

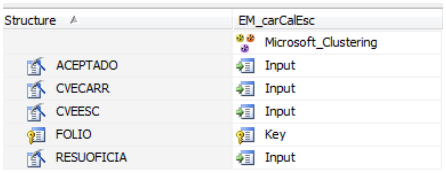


FIGURA 4:8 PANTALLA CON DATOS DE ENTRADA PARA EL EXPERIMENTO 1

4.2.1.1 Diagrama de cluster 1

La Figura 4:9 muestra seis clusters, el que se presenta en color azul obscuro es que tiene el porcentaje más alto de calificaciones. Las aristas con las que se une a los demás clusters indican la clusters con las que comparte similitudes. Entre más obscura se encuentre la arista la cercanía con el cluster es mayor, lo cual significa que comparten características similares.

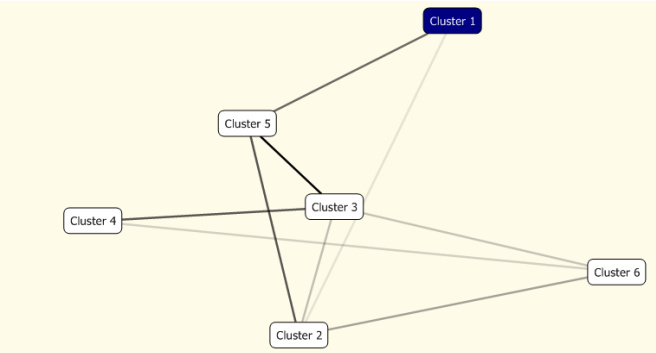


FIGURA 4:9 CLUSTER 1- CALIFICACIONES MÁS ALTAS

4.2.1.2 Características del cluster 1

En esta parte se presenta por porcentajes de probabilidad los elementos que conforman cada cluster. Se presentan los datos más significativos, es decir aquellos que contienen la mayoría de la información, por lo que las sumas posiblemente no sean igual al 100%.

En la Tabla 4:2 se presentan en color azul los atributos con mayor presencia en el cluster. El 46.68% indica que casi la mitad de los aspirantes en este cluster han obtenido entre 111 a 172 puntos, aunque el rango se abre desde 84 a 172. En lo que respecta al estado de nacimiento de los aspirantes, predomina el 22 que corresponde a Querétaro, con un 62.905% de probabilidad. De las escuelas el porcentaje mayor de probabilidad recae en la escuela 30 que es Prepa Norte UAQ, con un 12.205%.

TABLA 4-2 PROBABILIDADES DEL CLUSTER 1: CALIFICACIÓN, ESTADO, CLAVE DE ESCUELA

variables	Valores	probabilidad
RESUOFICIA	**111.0 - 172.0**	**46.688%**
RESUOFICIA	57.0 - 84.0	8.636%
RESUOFICIA	84.0 - 111.0	44.353%
EDONAC	11	8.723%
EDONAC	13	3.281%
EDONAC	15	3.241%
EDONAC	22	62.905%
EDONAC	9	14.046%
CVEESCORI	025	8.083%
CVEESCORI	029	7.683%
CVEESCORI	030	12.205%
CVEESCORI	031	10.684%

4.2.1.3 Diagrama del cluster 3

El cluster 3 tiene a los alumnos con calificaciones más bajas. Cumplen con ser puntuaciones que se encuentran en el rango de 57 a 111. (Figura 4:10)

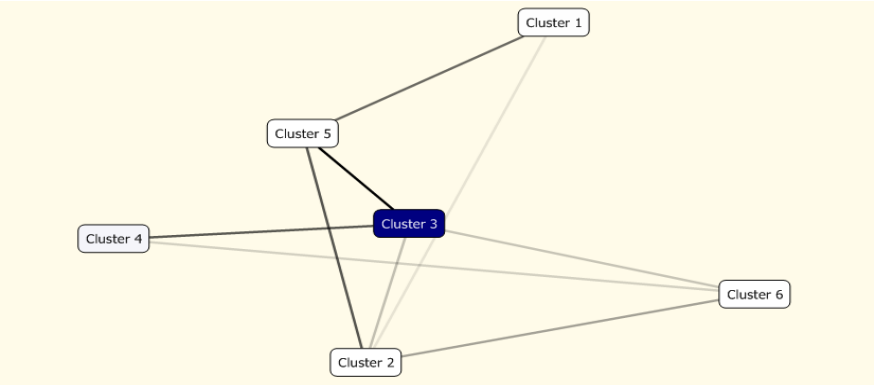


FIGURA 4:10 DIAGRAMA DEL CLUSTERS- CALIFICACIONES MÁS BAJAS

4.2.1.4 Características del cluster 3

TABLA 4-3 PROBABILIDADES DEL CLUSTER 3: CALIFICACIÓN, ESTADO, CLAVE DE ESCUELA

variables	valores	probabilidad
RESUOFICIA	57.0 - 84.0	98.842%
EDONAC	22	67.442%
EDONAC	9	12.474%
CVEESCORI	004	6.977%
CVEESCORI	025	8.879%
CVEESCORI	027	5.708%
CVEESCORI	029	6.342%
CVEESCORI	030	5.708%
CVEESCORI	031	5.708%

En la Tabla 4-3 se aprecia que el rango de calificaciones en este cluster va desde 57 a 84 puntos. Y el estado de nacimiento que predomina es también Querétaro. En este cluster las escuelas son muy diversas, las que obtienen las probabilidades más altas son 25 que corresponde a Colegio de Bachilleres Satélite, con 8.87% y al 29 con el 6.342 que es CBTIS 118.

4.2.1.5 Otros Clusters

Entre los cluster con calificaciones promedio tenemos al 1, 2 y 5. Como se puede apreciar en la Figura 4:11.

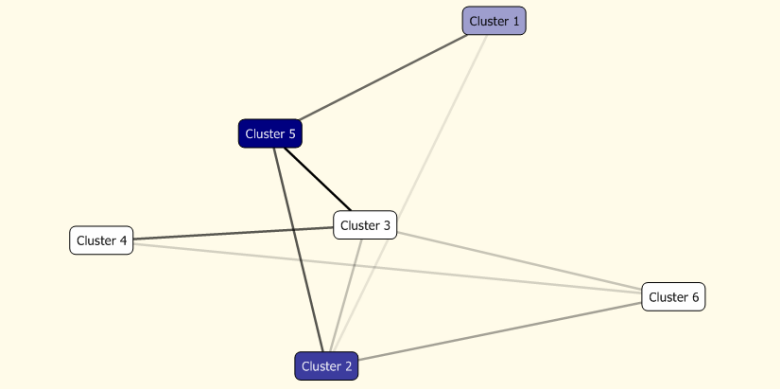


FIGURA 4:11 DIAGRAMA DEL CLUSTERS- CALIFICACIONES PROMEDIO

4.2.2 Análisis Experimental 2: resultados obtenidos por los aspirantes, folio del alumno y aceptado o no

Entrada: folio (identificador único del alumno), aceptado sí o no, resultado oficial.

TABLA 4-4 DATOS DE ENTRADA ANÁLISIS EXPERIMENTAL 2 CON EL ALGORITMO EM

Atributos	**FolioCalAceptado_EM**
Algoritmo	Microsoft_Clustering: EM
ACEPTADO	Input
FOLIO	Key
RESUOFICIA	Input

En la Tabla 4:4 se presentan los atributos de entrada para el presente análisis, así como el rol que ejerce cada uno de ellos. Por ejemplo, el atributo *folio* es utilizado como llave primaria, el atributo *aceptado* es entrada de datos para el análisis y el *resultado oficial* (resuoficia) es también entrada de datos.

4.2.2.1 Diagrama de clusters

El grafo que se presenta en la Figura 4:12 presenta una vista de todos los clusters, lo que la herramienta menciona como la población completa.

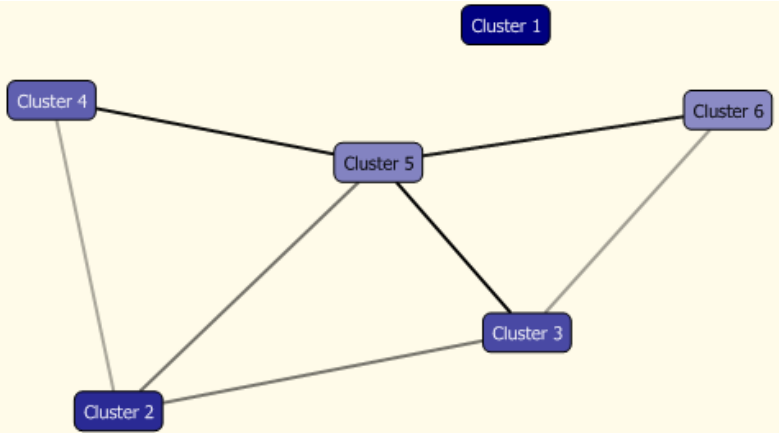


FIGURA 4:12 DIAGRAMA DE CLUSTERS PARA ASPIRANTES ACEPTADOS Y NO ACEPTADOS POR FOLIO Y CALIFICACIÓN

4.2.2.2 Características del cluster 1 Aspirantes aceptados

En la Tabla 4:5 se muestran por variables las probabilidades. Se observa que el cluster está compuesto en un 99.99% por aspirantes aceptados, cuyas calificaciones se encuentran entre 56.3 a 172. Es interesante comentar que el cluster 1 es el único que tiene a los alumnos aceptados.

TABLA 4-5 CARACTERÍSTICAS DEL CLUSTER 1: ASPIRANTES ACEPTADOS

variables	valores	probabilidad
ACEPTADO	S	99.998%
RESUOFICIA	111.3 - 172.0	55.020%
RESUOFICIA	83.8 - 111.3	40.773%
RESUOFICIA	56.3 - 83.8	4.132%

4.2.2.3 Características del cluster 5 Aspirantes no aceptados

En la Tabla 4;6 se muestran que las calificaciones de los aspirantes no aceptados que caen en éste cluster van desde 0 hasta 56.3.

TABLA 4-6 CARACTERÍSTICAS DEL CLUSTER 5: ASPIRANTES NO ACEPTADOS

variables	valores	probabilidad
ACEPTADO	N	99.586%
RESUOFICIA	0.0 - 56.3	50.000%

4.2.2.3.1 Comparación entre el cluster 1 y el 5

TABLA 4-7 COMPARACIONES ENTRE LOS CLUSTERS 1 Y 5

variables	**valores**	**Cluster 1**	**Cluster 5**
RESUOFICIA	0.0		100.000
RESUOFICIA	0.0 - 172.0	100.000	
ACEPTADO	N		98.547
ACEPTADO	S	98.547	

4.3 Caso Experimental utilizando k-medias

4.3.1 Análisis Experimental 1: resultados obtenidos por los aspirantes, plan de estudio y resultado oficial

Entrada: aceptado, clave carrera (plan de estudios), folio (identificador único del alumno), resultado oficial.

4.3.1.1 Diagrama del cluster con calificaciones altas

En este caso experimental los clusters que se forman se muestran en a Figura 4:13. Se muestran que el cluster 4 y el cluster 1 tienen las calificaciones más altas, por lo que son los que se analizaran.

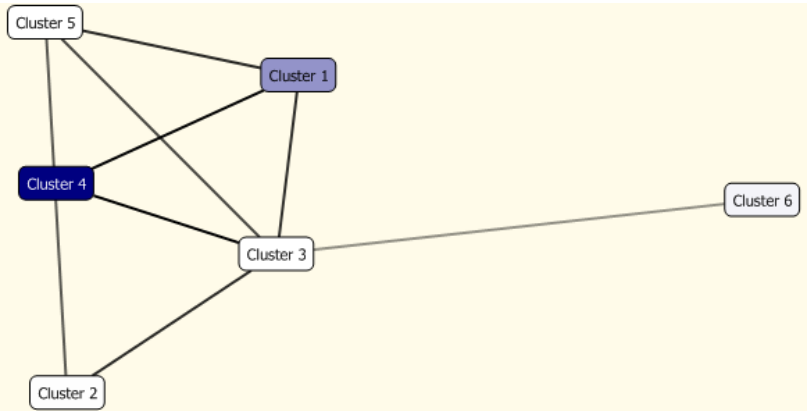


FIGURA 4:13 DIAGRAMA DE CLUSTERS CON CALIFICACIONES ALTAS, UTILIZANDO K-MEDIAS

4.3.1.2 Características del cluster 4

Se seleccionó primero éste grupo debido a que es el que muestra las calificaciones más altas. En la Tabla 10 se muestra que el rango de calificaciones es de 111 a 177.3. El estado de nacimiento tiene un rango de 1 a 14 y la escuela que se encuentra con más frecuencia en éste cluster es la 31 que corresponde a Prepa Sur UAQ.

Las conclusiones para éste análisis es que la escuela de procedencia de la cual los alumnos aspirantes a la Facultad de Informática alcanzan mejores

puntuaciones es la Prepa Sur UAQ, obteniendo entre 111 a 177.3 puntos. El rango de los estados de nacimiento de los alumnos es amplio y no podríamos identificar claramente uno predominante.

TABLA 4-8 CARACTERÍSTICAS DEL CLUSTER 4, CALIFICACIONES MÁS ALTAS

variables	valores	probabilidad
RESUOFICIA	111.0 - 177.3	100.000%
EDONAC	1.7 - 14.7	95.652%
CVEESCORI	031	21.739%
CVEESCORI	415	6.522%
CVEESCORI	030	6.522%
CVEESCORI	411	6.522%

4.3.1.3 Características del cluster 1

La Tabla 4:8 corresponde al cluster 1 que también presenta algunas calificaciones altas, pero el rango completo es mucho más grande y va desde 83.5 a 177.3, predominando con 41.66 % las calificaciones de 83.5 a 111.

El estado de nacimiento también muestra un rango amplio desde 1.7 a 33, lo cual no permite identificar un dominante.

Las claves de las escuelas con mayor probabilidad dentro del cluster son la 31 que corresponde a Prepa Sur UAQ, así como la 30 que corresponde la Prepa Norte UAQ siendo ésta última la que tiene la probabilidad más alta.

TABLA 4-9 CARACTERÍSTICAS DEL CLUSTER 1 UTILIZANDO K-MEDIAS

variables	Valores	probabilidad
RESUOFICIA	83.5 - 111.0	41.666%
RESUOFICIA	111.0 - 177.3	34.915%
EDONAC	1.7 - 14.7	28.269%
EDONAC	14.7 - 18.5	24.889%
EDONAC	18.5 - 22.3	23.552%
EDONAC	22.3 - 33.0	22.455%
CVEESCORI	030	11.568%
CVEESCORI	031	9.646%

4.3.2 Análisis Experimental 2: resultados obtenidos por los aspirantes, folio del alumno y aceptado o no

Entrada: folio (identificador único del alumno), aceptado sí o no, resultado oficial.

La Tabla 4:10 presenta los datos de entrada o atributos, así como el rol que estarán ejerciendo. Es necesario, como en cualquier base de datos relacional, identificar una clave o llave primaria, en este caso es el folio del alumno.

TABLA 4-10 DATOS DE ENTRADA ANÁLISIS EXPERIMENTAL 2 CON EL ALGORITMO K-MEDIAS

Atributo	**FolioCalAceptado_k-medias**
	Microsoft_Clustering
ACEPTADO	Input
FOLIO	Key
RESUOFICIA	Input

4.3.2.1 Diagrama de los clusters

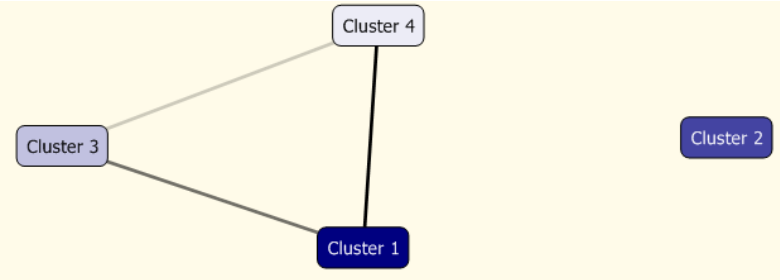
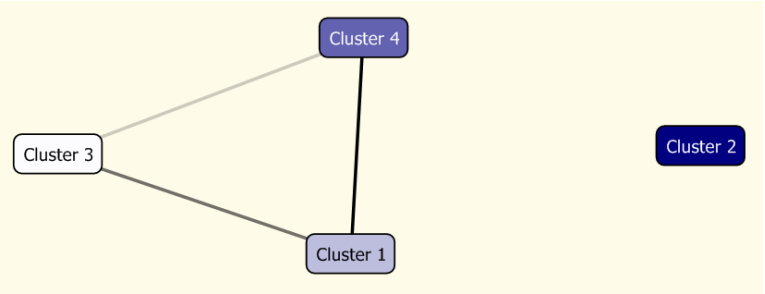


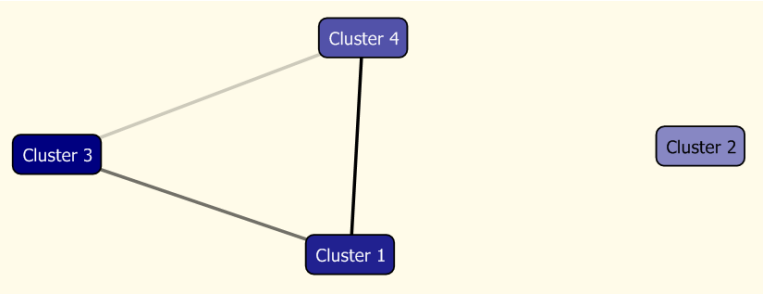
FIGURA 4:14 DIAGRAMA DE CLUSTERS ASPIRANTES ACEPTADOS O NO, UTILIZANDO K-MEDIAS

Cuando realizamos éste mismo experimento con el algoritmo EM obtuvimos un cluster agrupando a los aspirantes aceptados y otro con los aspirantes no aceptados. En el caso de k-medias el comportamiento es diferente. Por ello, se presentan los diagramas asociados.



En la Figura 4:15 se visualizan tres clusters con población de aspirantes aceptados en un porcentaje alto. El que predomina es el cluster 2 y le sigue el 4.

FIGURA 4:15 DIAGRAMA DE CLUSTERS CON ALUMNOS, ACEPTADOS K-MEDIAS



En la Figura 4:16 se observan cuatro cluster con aspirantes no aceptados, predominando el cluster 3 y el 1.

FIGURA 4:16 DIAGRAMA DE CLUSTERS CON ALUMNOS NO ACEPTADOS, K.MEDIAS

4.3.2.2 Características de los clusters con predominio de alumnos aceptados.

4.3.2.2.1 Cluster 2

TABLA 4-11 CARACTERÍSTICAS DEL CLUSTER 2 DONDE EL PORCENTAJE MAYOR CORRESPONDE A ASPIRANTES ACEPTADOS

variables	valores	probabilidad
RESUOFICIA	111.3 - 172.0	78.481%
ACEPTADO	S	53.191%
ACEPTADO	N	46.809%
RESUOFICIA	83.8 - 111.3	21.161%

La Tabla 4:11 nos presenta un porcentaje alto de alumnos aceptados, sin embargo de igual manera el porcentaje de alumnos no aceptados es alto. Lo que se podría tomar como punto representativo del cluster es el porcentaje de probabilidad que tienen de los alumnos agrupados en el cluster 2, es el rango de calificaciones que va desde 111 a 172, lo cual nos indican calificaciones altas.

4.3.2.2.2 Cluster 4

TABLA 4-12 CARACTERÍSTICAS DEL CLUSTER 4

Variables	Valores	probabilidad
RESUOFICIA	83.8 - 111.3	100.000%
ACEPTADO	N	67.320%
ACEPTADO	S	32.680%

4.3.2.3 Características de los clusters con predominio de alumnos no aceptados

4.3.2.3.1 Cluster 3

TABLA 4-13 CARACTERÍSTICAS CLUSTER 3 CON ALUMNOS NO ACEPTADOS

Variables	Valores	Probabilidad
ACEPTADO	N	99.582%
RESUOFICIA	0.0 - 56.3	50.000%

La Tabla 4:13 corresponde al cluster 3 y de manera clara podemos ver que son alumnos no aceptados, debido a que los resultados van desde 0 hasta 56-3, con una probabilidad del 50%.

4.3.2.3.2 Cluster 1

TABLA 4-14 CARACTERÍSTICAS DEL CLUSTER 1 ALUMNOS NO ACEPTADOS

Variables	Valores	Probabilidad
ACEPTADO	N	86.432%
RESUOFICIA	56.3 - 83.8	52.566%
RESUOFICIA	83.8 - 111.3	28.138%
RESUOFICIA	0.0 - 56.3	16.888%
ACEPTADO	S	13.568%
RESUOFICIA	111.3 - 172.0	2.408%

En la Tabla 4:14, los alumnos no aceptados tienen un 86.4% de probabilidad de pertenecer a éste cluster y los aceptados solamente un 13.5%.

4.4 Análisis de Resultados

Una vez que los modelos de DM se tienen listos, se puede llevar acabo diferentes tareas. Algunas tareas comunes que se pueden realizar: son el modelo para crear predicciones que puedan utilizarse para tomar decisiones, incrustar la funcionalidad de minería de datos directamente en una aplicación y crear un informe que permita a los usuarios realizar consultas directamente en un modelo de minería de datos existente. La actualización del modelo forma parte de la estrategia de implementación. A medida que la organización recibe más datos, debe volver a procesar los modelos para mejorar así su eficacia.

El hecho de que la herramienta permita varios algoritmos nos permite tener resultados complementarios distintos. Por lo que un análisis completo de BI sería utilizando además de los algoritmos de segmentación (que dividen los datos en grupos, o clústeres, de elementos que tienen propiedades similares. Un ejemplo de algoritmo de segmentación es el Algoritmo de clustering como los que utilizamos en el presente proyecto), algoritmos de clasificación (que predicen una

o más variables discretas basándose en otros atributos del conjunto de datos como árboles de decisión), algoritmos de asociación (que buscan correlaciones entre diferentes atributos de un conjunto de datos.), de algoritmos de secuencia (que resumen secuencias o episodios frecuentes en los datos), por nombrar algunos conocidos.

El algoritmo de EM nos arrojo las probabilidades que tienen los datos de caer en los seis diferentes clusters que se generaron. Definitivamente la información obtenida debe ser complementada con la que se pueda obtener de utilizar otras técnicas de BI.

.

5 Caso de estudio Agrupamiento de Entradas en una Matriz Binaria.

En este capítulo se utiliza una técnica de clustering que permite encontrar grupos en un espacio n-dimensional representado en una matriz con valores binarios. El caso de estudio es orientado a ubicar a los maestros que dentro de una Facultad tienen intereses en materias comunes de manera en que se pudiera lograr trabajo colegiado. El análisis experimental ésta basado en una matriz de entradas binarias en la cual interesa encontrar los grupos de datos con valor de uno, consideramos que el valor de uno estará presente en las materias que le interesan al maestro.

Suponga un matriz de datos $A = (a_{ij})$ de dimensiones $m \times n$. Donde a_{ij} mide la cercanía entre el renglón i y la columna j de la matriz. Si i y j son índices de renglones y columnas entonces, el renglón i podría ser una técnica de mercadotecnia y la columna j una aplicación para cada técnica que ha sido utilizada con éxito.

El objetivo es permutar los renglones y las columnas de A para que las relaciones entre los subconjuntos de renglones y columnas queden de una manera más clara. Por ejemplo la matriz de la Figura 5:1a no presenta ninguna relación fuerte, sin embargo la Figura 5:1b después de las permutaciones entre renglones y columnas, los renglones 1,3 y 5 son relacionados con la columna 1 y 3, y los renglones 2 y 4 son similares así como las columnas 2 y 4. La Figura 5:1B representa la mejor agrupación.

$$\begin{matrix}\mathbf{1}\\ \mathbf{2}\\ \mathbf{3}\\ \mathbf{4}\\ \mathbf{5}\end{matrix}\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} \end{bmatrix} \qquad \begin{matrix}\mathbf{1}\\ \mathbf{3}\\ \mathbf{5}\\ \mathbf{2}\\ \mathbf{4}\end{matrix}\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ \mathbf{1} & \mathbf{3} & \mathbf{2} & \mathbf{4} \end{bmatrix}$$

FIGURA 5:1A *FIGURA 5:1 B*

Ahora el problema es optimizar estas permutaciones, para lograrlo nos apoyaremos en el problema de maximización TSP por sus siglas en inglés de Travelling Salesman Problem, con una variante que lo convierte en el problema de ClusterTSP (Sharlee y Zhang 2004), computacionalmente equivalente.

El problema de Agrupamiento de Entradas (AE) en una matriz binaria es un problema combinatorio donde dada una matriz *A* de dimensiones *m* x *n* con entradas binarias, se desea agrupar los elementos con valor uno, para ello se permutan las columnas hasta lograr maximizar la función objetivo. La función objetivo es definida como se muestra a continuación:

$$\sum_{i=1}^{m-1}\sum_{j=1}^{n} a_{ij}\ a_{i+1,j} + \sum_{j=1}^{n-1}\sum_{i=1}^{m} a_{ij}\ a_{i,j+1} \quad (8)$$

La función objetivo consiste en la sumatoria de los productos escalares de los renglones y columnas contiguas, que son representados por el grado de entradas. Si cada renglón y cada columna es comparada con otras del vecindario entonces la función objetivo es pequeña, donde los renglones contiguos y las columnas son lexicográficamente cercanas (es decir que los caracteres que se analizan sean semejantes, en el caso del presente proyecto se tratan de agrupar los unos), el valor de la función objetivo es más grande. Así, lo que debemos tener es una matriz cuya función objetivo sea el máximo. Entonces la matriz presentará las entradas que pueden ser agrupadas.

5.1 TSP

El problema de TSP consiste en encontrar la ruta óptima a seguir para que el agente visite todas las ciudades y regrese a su punto de partida. Es decir, suponga que el punto inicial es v_1 y se requiere visitar v_2; v_3, v_4, v_5, v_6, v_7 y regresar a v_1. ¿Cuál sería la ruta óptima? (Figura 5:1).

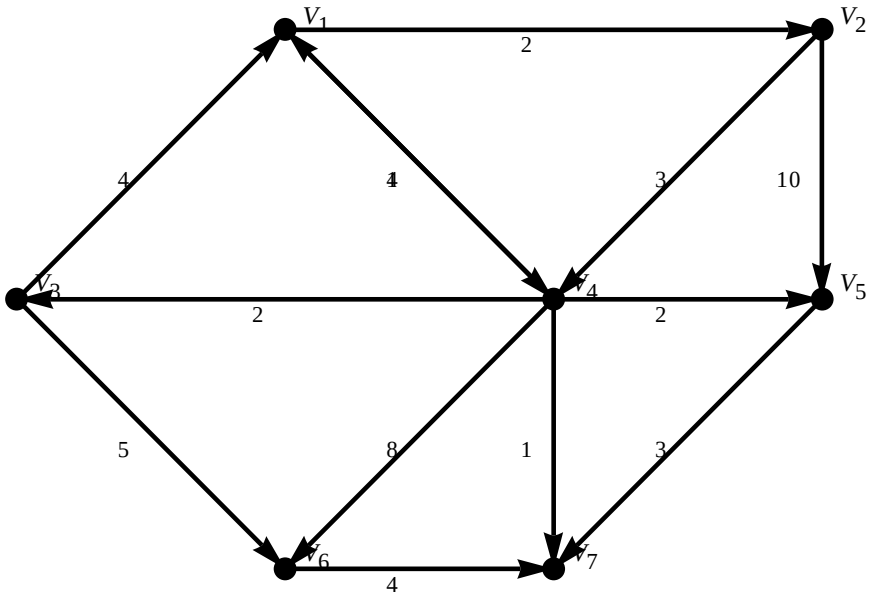


FIGURA 5:1 EJEMPLO DEL PROBLEMA TSP

A pesar de la aparente sencillez del planteamiento, el TSP es uno de los problemas más complejos para resolver. De manera formal, sean n ciudades de un territorio, el objetivo es encontrar una ruta que comenzando y terminando en una ciudad concreta, pase una sola vez por cada una de las ciudades y minimice la distancia recorrida por el viajante. Donde la distancia entre cada ciudad está dada por la matriz de dimensiones *n x n* donde *d(x,y)* representa la distancia que hay entre la ciudad x y la ciudad y.

Una solución no eficiente es recorrer todas las rutas posibles y así encontrar la que sea mejor utilizando menor distancia, sin embargo esta solución sería agotadora. Considerando que el problema reside en el número de posibles combinaciones dadas (N!) por el factorial del número de ciudades. El problema de TSP se considera como un problema computacionalmente intratable NP Completo.

Antes de continuar, se considera necesaria una breve explicación de la función que estaremos utilizando para encontrar los clusters.

5.2 La función FindClusters

Esta función se basa en algoritmos que por medio de calcular las distancias entre los elementos encuentra aquellos que por su cercanía forman grupos. Aquellos elementos que son idénticos tendrán una distancia igual a cero y los otros tendrán una distancia positiva. La sintaxis que utiliza es: **FindClusters[datos]** donde datos son la lista de elementos que se desean agrupar. Por defecto la función busca cuatro grupos, sin embargo es posible indicar la cantidad de clusters que se desean encontrar **FindClusters[datos,n]**. Es posible buscar clusters en un número arbitrario de dimensiones, la visualización de más de dos o tres dimensiones es complicada. Mathematica permite especificar incluso la función de distancia que se aplica para localizar los clusters, por defecto utiliza el algoritmo de vecino cercano (**Nearest**). La forma general para utilizar las funciones de distancias es:

FindClusters[{e_1,e_2,...},DistanceFunction->f]

La herramienta permite utilizar varios algoritmos estaremos utilizando el de distancia Euclidiana.

Es posible también especificar el algoritmo de clustering a utilizar, para ello se utilizará la siguiente sintaxis:

FindClusters[{e_1,e_2,...},Method→f}]

En el presente caso de estudio se estará trabajando con el método **Optimize** que permite encontrar clusters optimizados.

5.3 Valor de la función objetivo (OBV)

El valor de la función objetivo para una matriz A dada se puede calcular usando la función **objFunctionValue**, como sigue:

$$\mathbf{objFunctionValue}[\mathbf{A\_}] := \mathbf{Module}\left[\left\{i, j, m = \mathbf{Length}[A], n = \mathbf{Length}\left[A_{[[1]]}\right], \mathbf{At} = \mathbf{Transpose}[A]\right\}, \sum_{i=1}^{m-1} A[[i]].A[[i+1]] + \sum_{j=1}^{n-1} \mathbf{At}[[j]].\mathbf{At}[[j+1]]\right]$$

Considere el caso de una institución educativa consistente en un grupo de 10 profesores y un conjunto de 20 materias asociadas a los programas académicos que la institución ofrece. Sea *A* la matriz de dimensiones 20x10 que representa una lista de 20 cursos y 10 profesores que imparten en su conjunto dichas materias. La entrada *i* de la matriz *A* consiste en una lista que representa el subconjunto de profesores que están dispuestos y cuentan con las credenciales suficientes para impartir el curso *i*. Representaremos por 0 el caso para el cual el curso i no puede ser impartido por cierto profesor, y por 1 cuando sea lo contrario. A efecto de considerar varias instancias a lo largo de este capítulo, la matriz *A* es nombrada **instance1**, como sigue:

$$\text{instance1} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

El valor de la función objetivo de ***instance1*** luce como se muestra en la Figura 5:2, donde los cuadros negros representan la entrada binaria 1 y los cuadros blancos la entrada 0:

{objFunctionValue[instance1], ArrayPlot[instance1]}

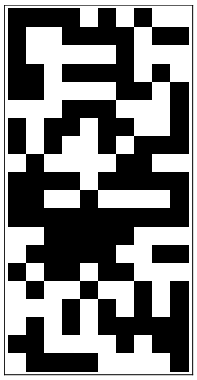


FIGURA 5:2 GRÁFICA INSTANCE1 OBV = 110

La pregunta ahora es ¿cómo puedo manipular las entradas de dicha matriz de modo que no destruya la información que ella representa, es decir, las preferencias de los profesores de impartir ciertas clases, y al mismo tiempo el valor de la función objetivo sea maximizado? En la siguiente sección

exploraremos una técnica basada en funciones de permutación de renglones y columnas.

5.4 Agrupando por renglones y columnas

Considere primeramente el enfoque de partición de las m entradas de la matriz A en k clusters de acuerdo al criterio de similaridad consistente en la distancia Euclidiana entre vectores *n*-dimensionales. Así, los vectores que habiten en un mismo cluster tendrán una distancia Euclidiana mínima y dos vectores que habiten en dos clusters diferentes tendrán una distancia mayor, aunque se puede dar el caso en el que los vectores se traslapen. En otras palabras, el método consiste en maximizar la distancia entre clusters al mismo tiempo que minimizamos las distancias entre los vectores dentro de los clusters. Para ello, se utilizara la función especializada **FindClusters** de *Mathematica*, para la cual, dada una matriz *A* y el número *k* de clusters produce la partición de la matriz *A* de acuerdo al criterio mencionado.

En virtud de que el número *k* de clusters impacta directamente en el valor de la función objetivo, es de considerar cual es el número más apropiado a definir. Existen dos valores extremos triviales para *k*. Estos son k=1 y k=m (en el caso de agrupar renglones) o k=n (para el caso en que agrupamos columnas). El primero contiene todos los elementos en el mismo cluster, mientras que el segundo construye un cluster para cada renglón o columna. El valor de k entonces, para fines prácticos, se define en función de un factor de granularidad g. Dicho factor de granularidad g se encuentra en el rango abierto <0,1>, y es utilizado como factor para determinar el número de clusters k. Así, k=g m (para determinar el número de clusters cuando se agrupan los renglones) o k=g n (para el caso en que se determina el numero de clusters para agrupar las columnas). La función **clustering1** recibe como datos de entrada la matriz A y el factor de granularidad g, y con ellos lleva a cabo el agrupamiento primeramente por renglones y enseguida por columnas, como sigue:

```
clustering1[A_, gf_] := Module[{permuteByRow, permuteByColumn,
m = Length[A],      n = Length[A[[1]]]},
permuteByRow = Flatten[FindClusters[A, Floor[gfm], Method
                -> "Optimize"], 1];
permuteByColumn =
Flatten[FindClusters[Transpose[permuteByRow], Floor[gfn], Method ->
Optimize, 1 //Transpose
{objFunctionValue[permuteByColumn], permuteByColumn}]
```

Considere el ejemplo dado en la sección anterior referente al caso titulado ***instance1***. En seguida se muestran tanto los valores de la función objetivo como la matriz *A* para la matriz original y la matriz producida por la función ***clustering1***:

```
{OFV, solution1} = clustering1[instance1, 0.5];
{objFunctionValue[instance1],
ArrayPlot[instance1],
OFV, ArrayPlot[solution1]}
```

Función objetivo = 110

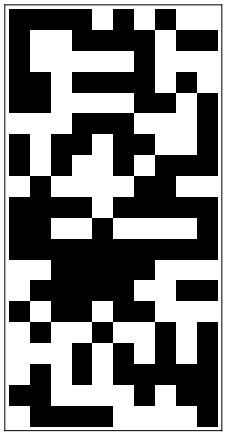


FIGURA 5:3 SIN APLICAR LA FUNCIÓN DE CLUSTERING

Función objetivo = 139

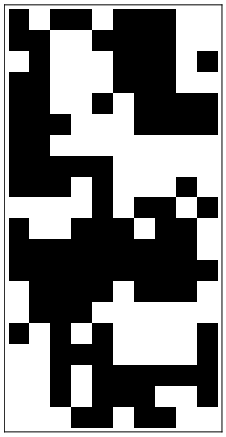


FIGURA 5:4 APLICANDO LA FUNCIÓN DE CLUSTERING

El valor de la función objetivo creció de 110 a 139. La Figura 5:4 muestra un agrupamiento más compacto en comparación con la Figura 5:3 asociada a la matriz original. Hemos encontrado y verificado experimentalmente que si la función **clustering1** lleva a cabo primeramente el agrupamiento por columnas y después por renglones, los resultados son exactamente los mismos. La información no se pierde.

Hemos mencionado que el rango del factor de granularidad define el numero k de clusters. Enseguida mostramos como g afecta el valor de la función objetivo para nuestro ejemplo dado en **instance1**. En la Figura 5:5 se presenta la gráfica perfil del valor de la función objetivo de acuerdo a diferentes valores de g.

$$\begin{aligned}&\mathbf{ListPlot}[\{\{\#, \mathbf{First}[\mathbf{clustering1}[\mathbf{instance1}, \#]]\}\&\\&\qquad /@Table[gf, \{gf, 0.1,1,0.01\}],\\&\quad \{\#, \mathbf{First}[\mathbf{clustering2}[\mathbf{instance1}, \#]]\}\&\\&\qquad\qquad /@Table[gf, \{gf, 0.1,1,0.01\}]\},\\&\quad \mathbf{Joined} \rightarrow \mathbf{True},\\&\quad \mathbf{Frame} \rightarrow \mathbf{True},\\&\quad \mathbf{Axes} \rightarrow \mathbf{True},\\&\quad \mathbf{GridLines} \rightarrow \{\mathbf{Table}[\mathbf{gf}, \{\mathbf{gf}, \mathbf{0.1}, \mathbf{1}, \mathbf{0.1}\}],\\&\quad \mathbf{Table}[\mathbf{gf}, \{\mathbf{gf}, \mathbf{100}, \mathbf{140}, \mathbf{10}\}]\},\end{aligned}$$

PlotRange → {{0.1, 1}, {100, 140}}, AspectRatio → 0.75,
FrameStyle → Thick,
Frame → True, Background → LightBlue,
FrameLabel → {Granurality Factor, OFV, Matrix A_{mxn}},
PlotLabel
→ Objective function value in terms of the granularity factor
GridLines → Automatic]

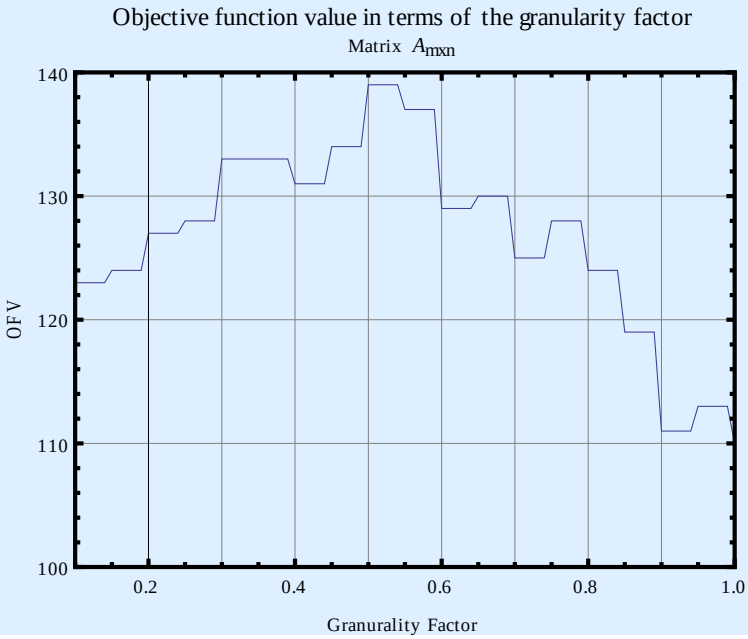


FIGURA 5:5 PERFIL DEL VALOR DE LA FUNCIÓN OBJETIVO DE ACUERDO A DIFERENTES VALORES DE G.

De la Figura 5:5 se puede establecer que la función objetivo crece en la primera parte hasta un valor central y después decrece. ¿Cuál es el valor de g que produce el máximo valor de la función objetivo y por ende el número más conveniente de k? Hasta el momento, y de acuerdo a la Figura 5:5, el mejor valor de g corresponde a 0.5. Esto significa que los renglones fueron particionados en 10 (=20 x 0.5) y las columnas en 5(=10 x 0.5) clusters. Enseguida extendemos nuestro experimento considerando 15 matrices de dimensiones 200x200, y determinamos para cada una de ellas el valor óptimo de g, y así k. La lista producida por la función **clustering1** para cada una de las 15 matrices consiste en el valor óptimo de g y el valor de la función objetivo asociado.

```
m = n = 200; instanceSet =
    Table[Table[  Table[Random[Integer, {0, 1}],{m}], {n}], {15}];
   solutionSet =
            Map[Table[{g, clustering1[#, g] // First},
                {g, 0.1, 1, 0.1}] &, instanceSet];
   max =  Map[Last[Sort[#, Last[#1] <= Last[#2] &]] &, solutionSet]
{{0.30000000000000004, 21311},
{0.30000000000000004, 21129},
{0.30000000000000004, 20946},
{0.4, 21780},
{0.30000000000000004, 21204},
{0.30000000000000004, 21146},
{0.4, 21145},
{0.30000000000000004, 21638},
{0.4, 21177},
{0.4, 21415},
{0.4, 21562},
{0.30000000000000004, 21439},
{0.30000000000000004, 21434},
{0.30000000000000004, 21680},
{0.4, 21218}}
```

Podríamos establecer que el valor de g se encuentra entre 0.3 y 0.4 para los 15 casos muestra.

A continuación, se muestra el perfil de cada una de las funciones objetivos. Observe el punto máximo de cada una de las gráficas, el cual ocurre en el rango entre 0.3 y 0.4. (Figura 5:6)

```
ListPlot[solutionSet, Joined → True, Frame → True, Axes → True,
GridLines → {Table[gf, {gf, 0., 1, 0.1}],
Table[gf, {gf, 20000, 22000, 100}]},
PlotRange → {{0, 1},
```

```
{20000, 22000}},
AspectRatio → 0.75,
FrameStyle → Thick, Frame → True,
Background → LightBlue,
FrameLabel {"Granurality Factor",
"OFV", "15 random matrices"A"20""x20"},
PlotLabel → Objective function value in terms of the granularity factor
GridLines → Automatic]
```

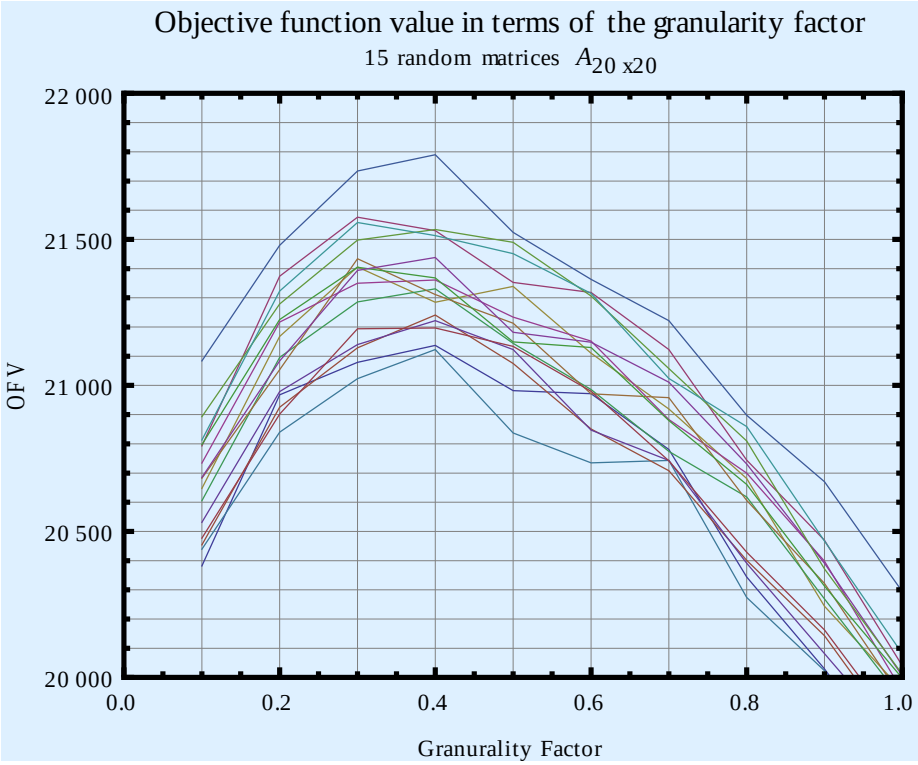


FIGURA 5:6 VALOR DE LA FUNCIÓN OBJETIVO EN TÉRMINOS DEL FACTOR DE GRANULARIDAD

5.5 Reduciendo la instancia de clustering al problema de TSP en un espacio multidimensional

El segundo método a seguir para optimizar la función objetivo planteado al inicio de este capítulo, consiste en el replanteamiento de nuestro problema original de Agrupamiento de Entradas en una Matriz Binaria (AE) en términos de un problema clásico de la computación llamado TSP.

El método consiste en la reducción del problema AE de una matriz binaria al problema TSP, de la siguiente manera: a cada renglón *i* de la matriz *A* le corresponde el vértice *i* en el grafo completo K_m, y el valor del producto punto entre el vector *i* y *j* corresponde al peso de la arista *{i,j}* de K_m.

Es importante señalar que las funciones objetivo de los problemas AE y TSP son equivalentes. Sin embargo, el problema AE consiste en maximizar el valor de su función objetivo, mientras que el problema TSP consiste en minimizar su función objetivo.

A fin de poder establecer una relación directa entre las soluciones a ambos problemas, se asigna un factor de -1 a cada unos de los pesos de las aristas del grafo K_m. De esta manera, es interesante notar que la solución del problema TSP consistente en un ciclo Hamiltoniano (en un grafo es un camino, una sucesión de aristas adyacentes, que visita todos los vértices del grafo una sola vez) de longitud mínima constituye el ordenamiento de los m renglones de la matriz *A*, y viceversa. La reducción es repetida ahora en términos de columnas a la matriz que resulta de la reducción en términos de renglones.

A continuación la función ***clustering3*** lleva a cabo la reducción del problema AE al TSP, y produce una matriz cuyo valor de la función objetivo es maximizado.

```
clustering3[A_]:=Module[{m=Length[A],
        n=Length[Subscript[A, [[1]]]],d,graph,totalLength,
        tourByRow,tourByColumn,tourByRowPermuted,
```

```
            tourByColumnPermuted,temp},
    graph=Map[#→-A[[Subscript[#,
            [[1]]]]].A[[Subscript[#, [[2]]]]]&, Subsets[Range[m],{2}]];
    graph=Union[graph,Map[Reverse[First[#]]→ Last[#]&,graph]];
    d=SparseArray[graph,{m,m}];
    {totalLength,tourByRow}=FindShortestTour[Range[m],
            DistanceFunction→ (d[[#1,#2]]&)];
    tourByRowPermuted={};
    tourByRowPermuted=Flatten[Map[Append[tourByRowPermuted,
            A[[#]]]&,tourByRow],1];
    temp=Transpose[tourByRowPermuted];
    graph=Map[#→-temp[[Subscript[#, [[1]]]]].temp[[Subscript[#,
            [[2]]]]]&, Subsets[Range[n],{2}]];
    graph=Union[graph,Map[Reverse[First[#]]→ Last[#]&,graph]];
    d=SparseArray[graph,{n,n}];
    {totalLength,tourByColumn}=FindShortestTour[Range[n],
            DistanceFunction→ (d[[#1,#2]]&)];
    tourByColumnPermuted={};
    tourByColumnPermuted=Flatten[Map[Append[tourByColumnPermuted,
            temp[[#]]]&,tourByColumn],1];
    {objFunctionValue[tourByColumnPermuted],
            Transpose[tourByColumnPermuted]}]
```

Tomemos nuevamente nuestro ejemplo de la sección anterior correspondiente al caso dado por ***instance1***.

```
{OFV, solution1} =
      clustering3[instance1]; {OFV,  ArrayPlot[solution1, Mesh → False]}
```

Función objetivo =146

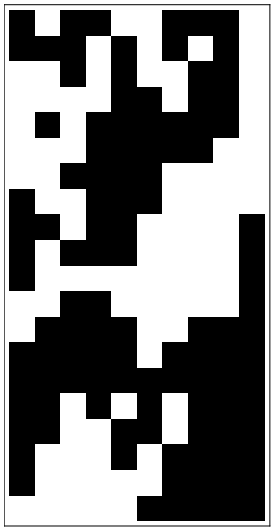


FIGURA 5:7 RESULTADO UTILIZANDO TSP

Como se puede observar para este caso particular el valor de la función objetivo crece de 110 asociada a la matriz original a 146, y más aun el valor de la función objetivo de 139 producida por la función **clustering1** es mejorada. Hemos probado experimentalmente que la reducción por columnas y después por renglones produce los mismos resultados en la función objetivo que cuando se reduce el problema AE al TSP primeramente por renglones y después por columnas.

A continuación presentamos a manera de resumen la comparación de los resultados de las funciones **clustering1** y **clustering 3**.

```
{OFV1,solution1}=clustering1[instance1,0.5];
{OFV3,solution3}=clustering3[instance1];

{objFunctionValue[instance1],

ArrayPlot[instance1,Mesh→False],

OFV1,ArrayPlot[solution1,Mesh→False],
OFV3,ArrayPlot[solution3,Mesh→False]}
```

Función objetivo = 110 inicial

Función objetivo=139 utilizando únicamente clustering

Función objetivo= 146 utilizando clustering y TSP

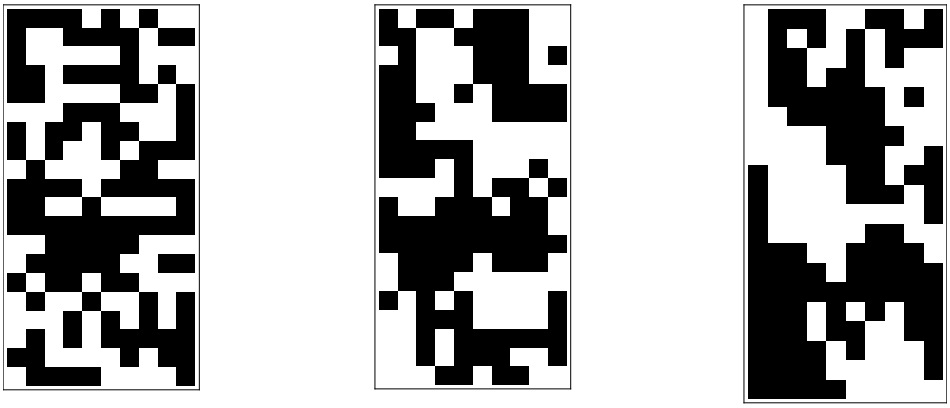


FIGURA 5:8 COMPARACIÓN ENTRE CLUSTERING1 Y CLUSTERING3

5.6 Análisis experimental del rango de aproximación

En esta sección se muestra experimentalmente el rango de aproximación del método basado en la reducción del problema AE al problema TSP, así como la eficiencia del algoritmo en términos de tiempos de ejecución como una función del tamaño de la matriz A. Considere la matriz A dada en ***instance2*** como una lista de listas, donde cada uno de los elementos corresponde a los renglones de A.

Es importante señalar que la matriz A en si corresponde a la manera optima de compactar las entradas de la matriz, de modo que el valor de la función objetivo corresponde al valor máximo de 96.

$$\textbf{instance2} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

{objFunctionValue[instance2], ArrayPlot[instance2]}

De la matriz A dada e ilustrada gráficamente en la Figura 5:8 se pueden construir tantas instancias como número de permutaciones entre renglones y columnas sea posible hacer. Tal numero de permutaciones por fila y por columna corresponde al producto *m! n!.* Para nuestro caso en ***instance2***, tal número es:

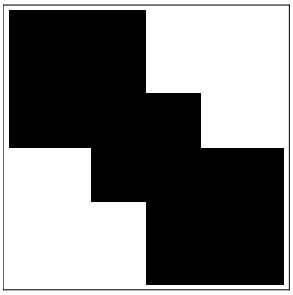


FIGURA 5:9 MATRIZ A

m = Length[instance2]; n = Length[First[instance2]]; m! n!

13168189440000

Este número representa el tamaño del espacio de instancias de matrices de dimensión 10x10 que poseen como solución óptima la matriz *A* dada en **instance2**. Este espacio de posibilidades puede entonces ser usado para probar nuestros algoritmos. Sin embargo, dado que el número de instancias es difícil de manejar, solo consideraremos un subconjunto de dichas instancias. Para definir tal subconjunto, primeramente implementamos la función **myShuffle** para la que dada una dimensión de la matriz y el número de elementos a permutar, se produce de manera aleatoria una función de permutación.

$$\mathbf{myShuffle}[\mathbf{n\_}, \mathbf{m\_}] := \mathbf{Module}[\{\mathbf{sol} = \{\},$$

$$\mathbf{elem} = \mathbf{Random}[\mathbf{Integer}, \{\mathbf{1}, \boldsymbol{n}\}]\},$$

$$\mathbf{While}[\mathbf{Length}[\mathbf{sol}] < \boldsymbol{m},$$

$$\mathbf{sol} = \mathbf{If}[\neg\mathbf{MemberQ}[\mathbf{sol}, \mathbf{elem}],$$

$$\mathbf{Append}[\mathbf{sol}, \mathbf{elem}], \mathbf{sol}];$$

$$\mathbf{elem} = \mathbf{Random}[\mathbf{Integer}, \{\mathbf{1}, \boldsymbol{n}\}]]; \mathbf{sol}]$$

Por ejemplo, considere el numero de renglones n=10 de **instance2**, para la cual queremos permutar solamente 5 de los renglones.

myShuffle[10, 5]

{7, 6, 2, 4, 8}

El resultado consiste en la permutación expresada en su formato de ciclo que significa que el renglón 7 va a la posición original del renglón 6, el renglón 6 a la posición 2, el 2 a la posición 4, y finalmente el 4 a la posición 8. Para nuestro caso en **instance2**, se genera aleatoriamente una permutación por renglones y después una permutación por columnas para generar una nueva matriz asignada a **instance2Shuffled**, como sigue:

```
instance2Shuffled=
     Transpose[Map[instance2[[#]]&,
               myShuffle[10,10]]];
     instance2Shuffled=Transpose[Map[instance2Shuffled[[#]]&,
               myShuffle[10,10]]];
     {ArrayPlot[instance2,Mesh→True],
     ArrayPlot[instance2Shuffled,Mesh→True]}
```

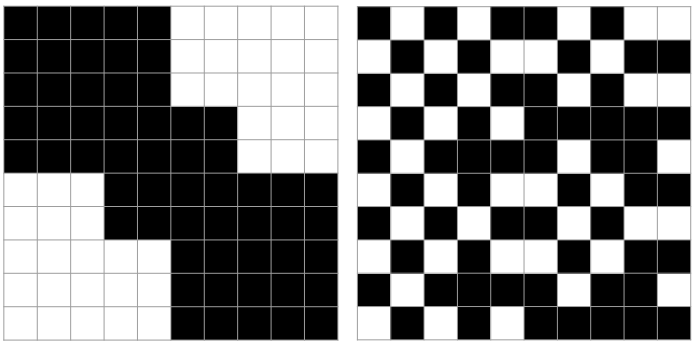


FIGURA 5:10 GRÁFICAS DE INSTANCE2SHUFFLED

A continuación se presentan los resultados consistentes en el valor de la función objetivo y la matriz en forma gráfica de la matriz en **instance2Shuffled**, el resultado producido por **clustering1**, y el resultado generado por **clustering3**.

{OFV1, solution1} = clustering1[instance2Shuffled, 0.45];

```
{OFV3, solution3} = clustering3[instance2Shuffled];
{objFunctionValue[instance2Shuffled],
 ArrayPlot[instance2Shuffled, Mesh → False],
OFV1, ArrayPlot[solution1,  Mesh → False],
OFV3,  ArrayPlot[solution3,       Mesh → False]}
```

Función objetivo = 36 Función objetivo = 86 Función objetivo = 96

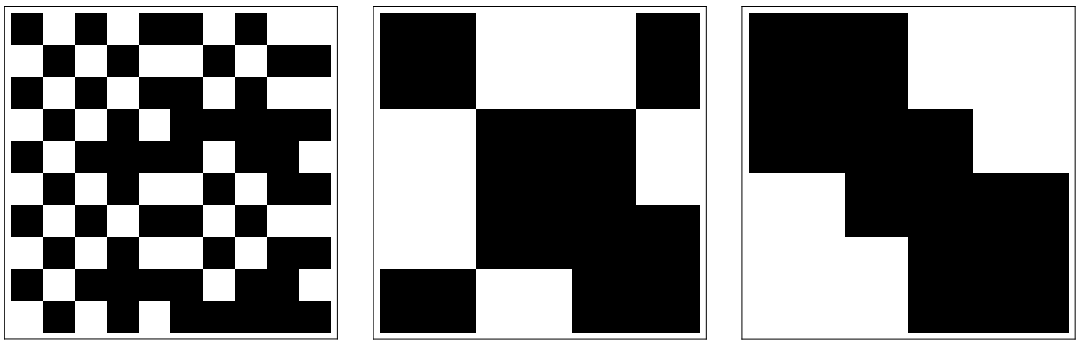


FIGURA 5:11 FORMA GRÁFICA DE LA MATRIZ INSTANCE2SHUFFLED

En este caso particular asociado a ***instance2Shuffled*** el segundo método basado en el algoritmo para TSP produce el valor óptimo de 96, el cual es mucho mejor que el resultado producido por el método basado en la función **FindClusters** provisto por *Mathematica* con valor de 86. Sin embargo, el método basado en el TSP no siempre produce el valor óptimo. Veamos cómo se comporta para un conjunto de 5040 matrices producidas a partir de la matriz en **instance2**. Generamos primeramente el mismo número de 5040 permutaciones de manera aleatoria.

```
permutations =
      Partition[#, 2, 1] & /@ (Append[#, First[#]] &
/@   Permutations[myShuffle[10, 7]])
Length[permutations]
```

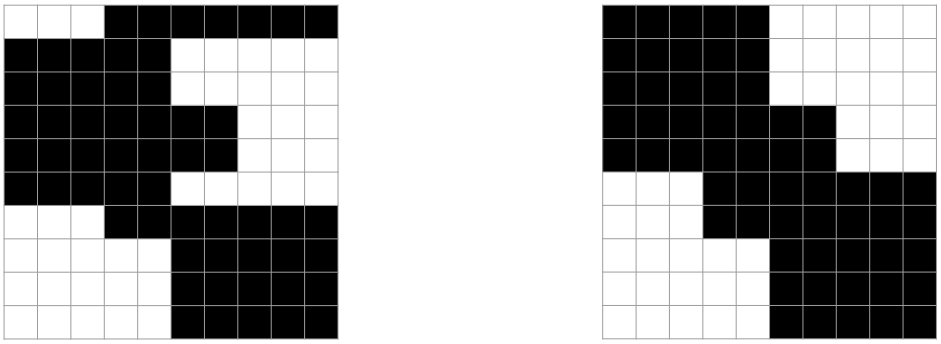
```
5040
```

Mediante la función ***mySwitch*** aplicamos la permutación *(i, 6, i),* esto es, el renglón i va a la posición original del renglón 6, y el renglón 6 va a la posición

original del renglón 1. Así que esta función es aplicada sistemáticamente a cada elemento de las 5040 permutaciones generadas.

```
mySwitch[i_, j_, instance_] :=
 Module[{instanceTemp = instance, temp},
 temp = instanceTemp[[i]];
 instanceTemp[[i]] = instanceTemp[[j]];
 instanceTemp[[j]] = temp;
 instanceTemp]

{ArrayPlot[mySwitch[1, 6, instance2], Mesh → True],
        ArrayPlot[instance2, Mesh → True]}
```



(a) (b)

FIGURA 5:12 INTERCAMBIANDO RENGLONES 1 Y 6

La Figura 5:12a es obtenida después de haber intercambiado los renglones 1 y 6 en la matriz de la Figura 5:12b. La función ***mySwitching*** toma una permutación y la aplica a la instancia dada, como sigue:

```
mySwitching[perm_,instance_]:=Module[{instanceNew=instance},
(instanceNew=mySwitch[Subscript[#, [[1]]],
Subscript[#, [[2]]],instanceNew])&/@perm;instanceNew]
```

Así, el conjunto de permutaciones es aplicado a la matriz de ***instance2***, produciendo así 5040 instancias de prueba cuya solución corresponde con la matriz ***instance2***.

```
instances = mySwitching[#, instance2] & /@ permutations;
```

Apliquemos ahora la función *clustering4* basada en el algoritmo para el TSP al conjunto de 5040 instancias, como sigue:

```
experiment1 = First[clustering4[#]] & /@ instances;
```

En la siguiente Figura se muestra que los valores de la función objetivo para el conjunto de instancias se encuentran en el rango desde 91 hasta el valor óptimo de 96.

```
ListPlot[experiment1,
AspectRatio → 0.75, FrameStyle → Thick,
Frame → True,
Background → LightBlue,
FrameLabel → {"Instance Number", "OFV", "Optimal OFV: 96"},
PlotLabel →  "Objective function value of 5040 10x10-
      matrices  from shuffling   instance2", GridLines → Automatic]
```

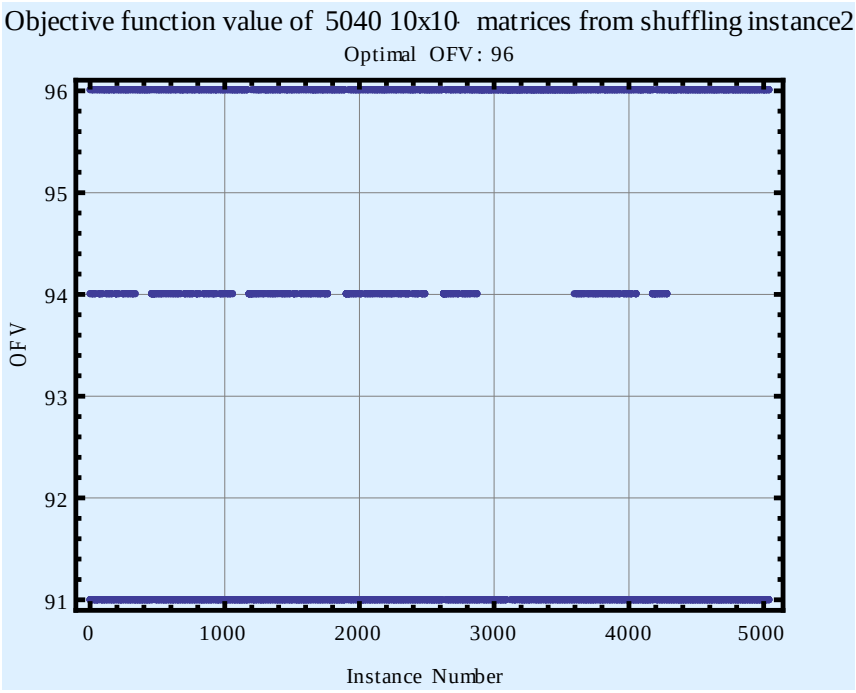


FIGURA 5:13 VALOR DE LA FUNCIÓN OBJETIVO DE 5040 EN UNA MATRIZ DE 10 X 10

El rango de aproximación promedio es del 97%, esto es, el valor promedio de la función objetivo es 0.97 x 96=93.12. Contraste este valor con el valor de la función objetivo optima de 96.

$$N\left[\frac{\text{Plus@@}\frac{\text{experiment1}}{96}}{\text{Length[experiment1]}}\right]$$

```
0.970734
```

Es posible también medir el tiempo de ejecución que el programa requiere como sigue:

```
experiment2 = First[Timing[clustering4[#]]] & /@ instances;
```

Gráficamente

```
ListPlot[experiment2, AspectRatio → 0.75, FrameStyle → Thick,
    Frame → True, Background → LightBlue,
    FrameLabel → {"Instance Number", "Running Time (sec.)",
      "For 5040 10x10-matrices from shuffling instance2"},
    PlotLabel → "Running Time of the TSP-based clustering function",
     GridLines → Automatic]
```

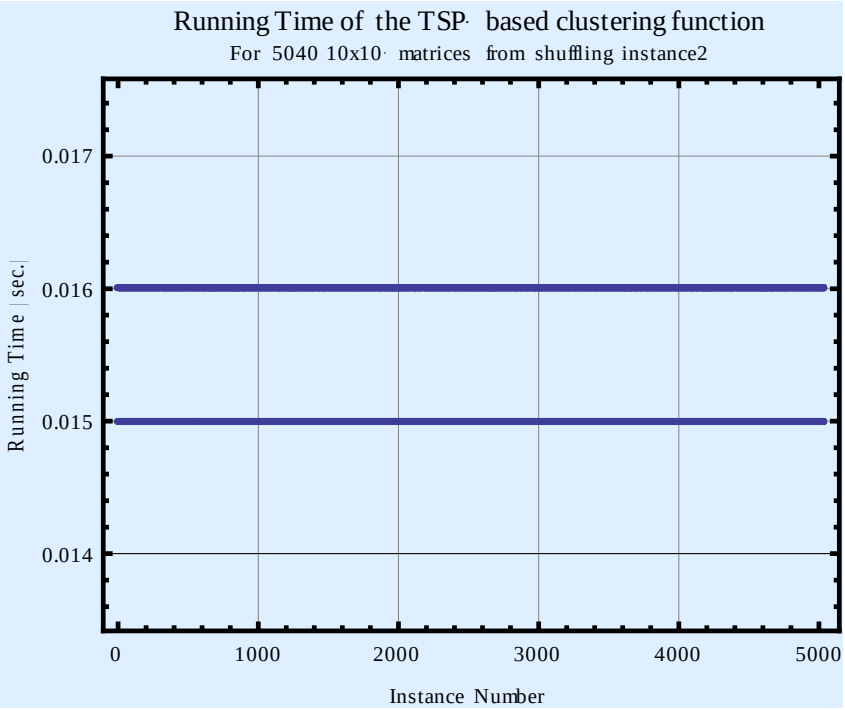


FIGURA 5:14 TIEMPOS DE EJECUCIÓN DE LA FUNCIÓN CLUSTERING TSP

El tiempo de ejecución para las 5040 instancias se encuentra en el rango de 15 a 16 mseg, en una computadora con procesador Intel Core 2 Duo a 2.60 GHz con 4.0 GB de memoria RAM.

Por lo que se refiere a la función ***clustering2*** de acuerdo al método basado en la función especializada **FindClusters** de *Mathematica*, los valores de la función objetivo para las 5040 instancias generadas, son como sigue:

```
experiment3 = First[clustering2[#, 0.5]] & /@ instances;

Length[experiment3]= 5040
```

Gráficamente:

```
ListPlot[experiment3, AspectRatio → 0.75,
FrameStyle → Thick,
Frame → True,
Background → LightBlue,
FrameLabel → {"Instance Number", "OFV", "Optimal OFV: 96"},
PlotLabel →  "Objective function value of 5040 10x10-matrices from shuffling \
instance2", GridLines → Automatic]
```

Si comparamos la Figura 5:15 con aquella que contiene los valores de la función objetivo utilizando el algoritmo para el TSP, es evidente que esta última produce valores de la función objetivo por debajo del rango entre 94 y 96 de aquella.

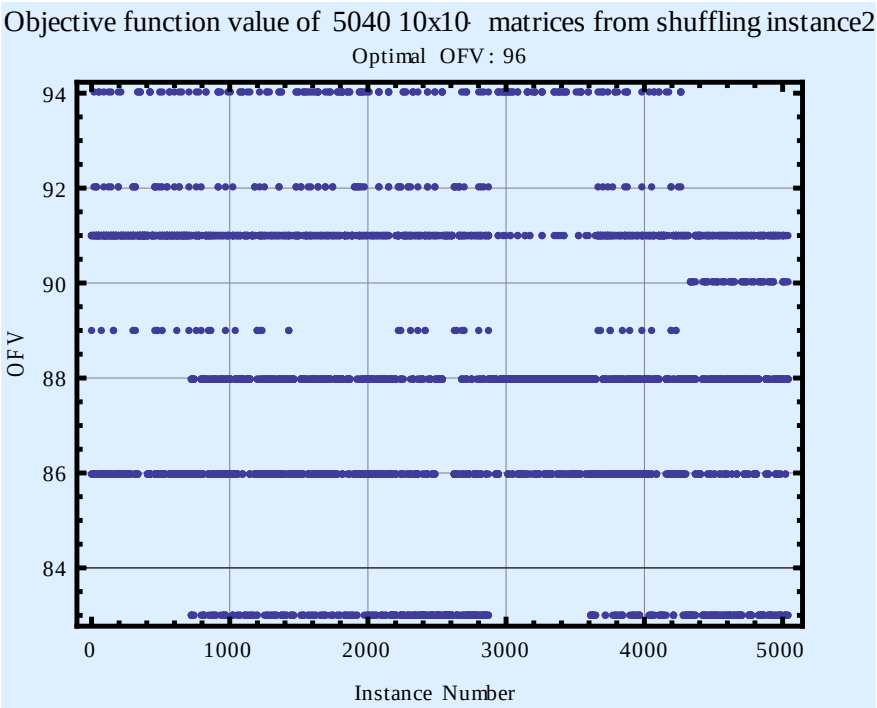


FIGURA 5:15 VALOR DE LA FUNCIÓN OBJETIVO DE 5040

Asimismo, podemos observar que el valor promedio del rango de aproximación de ambos métodos van desde 92% (el método basado en la función **FindClusters**) al 97% (el método basado en la reducción del problema al TSP). Con base en esto podríamos establecer que el último es mejor que el primero.

Podemos también determinar los tiempos de ejecución del algoritmo basado en **FindClusters** para las 5040 instancias como sigue:

```
experiment4 = First[Timing[clustering2[#, 0.5]]] & /@ instances;
```

Gráficamente:

ListPlot[experiment4, AspectRatio → 0.75, FrameStyle → Thick, Frame
→ True, Background → LightBlue, FrameLabel
→ {"Instance Number", "Running Time (sec.)", "For 5040 10x10
− matrices from shuffling instance2"}, PlotLabel
→ "Running Time of the clustering function using""FindClusters""function", GridLines
→ Automatic]

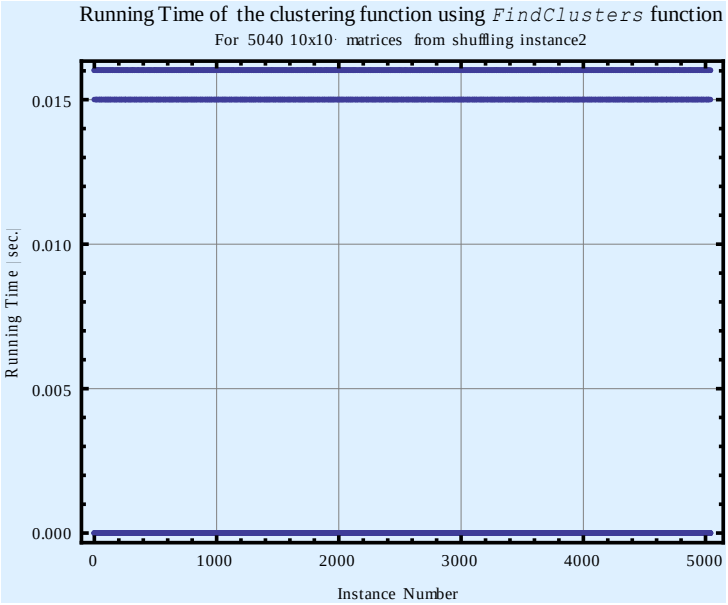


FIGURA 5:16 TIEMPOS DE EJECUCIÓN DEL ALGORITMO BASADO EN FINDCLUSTERS

Los tiempos de ejecución requeridos por el método basado en la función **FindCluster** de *Mathematica* son casi similares a aquellos requeridos por el método basado en algoritmo para el TSP, aunque para algunas instancias el tiempo es realmente casi 0. Los resultados indican entonces que al buscar métodos más eficientes se debe estar dispuesto a sacrificar optimalidad, y así el método basado en **FindClusters** es mejor, ya que es más rápido en general.

A continuación, se calcula vía análisis experimental la eficiencia del método basado en la reducción del problema AE al problema TSP usando la función clustering3. Considere primeramente el conjunto de instancias consistente en 15 matrices de dimensiones 20x20.

$$\boldsymbol{m = n = 20};$$

$$\mathbf{instanceSet} = \mathbf{Table}[\mathbf{Table}[\mathbf{Table}[\mathbf{Random}[\mathbf{Integer}, \{\mathbf{0}, \mathbf{1}\}], \{\boldsymbol{m}\}], \{\boldsymbol{n}\}], \{\mathbf{15}\}];$$

$$\mathbf{solutionSet} = \frac{\mathbf{Plus@@Map}[\mathbf{First}[\mathbf{Timing}[\mathbf{clustering4}[\#]]]\&, instanceSet]}{\mathbf{Length}[\mathbf{instanceSet}]}$$

El tiempo promedio de ejecución en segundos para las instancias de tamaño 20x20 es como sigue:

```
SolutionSet

0.0759333
```

Ahora procedemos con 15 instancias de dimensiones 30x30:

$$\boldsymbol{m = n = 30};$$

$$\mathbf{instanceSet} = \mathbf{Table}[\mathbf{Table}[\mathbf{Table}[\mathbf{Random}[\mathbf{Integer}, \{\mathbf{0}, \mathbf{1}\}], \{\boldsymbol{m}\}], \{\boldsymbol{n}\}], \{\mathbf{15}\}];$$

$$\mathbf{solutionSet} = \frac{\mathbf{Plus@@Map}[\mathbf{First}[\mathbf{Timing}[\mathbf{clustering4}[\#]]]\&, instanceSet]}{\mathbf{Length}[\mathbf{instanceSet}]}$$

El tiempo promedio de ejecución en segundos requerido es:

```
SolutionSet

0.189267
```

Para 15 matrices de dimensiones 40x40, su tiempo promedio es 0.394133 segundos, como sigue:

$$\boldsymbol{m = n = 40};$$

$$\mathbf{instanceSet} = \mathbf{Table}[\mathbf{Table}[\mathbf{Tabl}[\mathbf{Random}[\mathbf{Integer}, \{\mathbf{0}, \mathbf{1}\}], \{\boldsymbol{m}\}], \{\boldsymbol{n}\}], \{\mathbf{15}\}];$$

$$\mathbf{solutionSet} = \frac{\mathbf{Plus@@Map}\left[\mathbf{First}\left[\mathbf{Timing}\left[\mathbf{clustering4}[\#]\right]\right]\&, \mathbf{instanceSet}\right]}{\mathbf{Length}[\mathbf{instanceSet}]}$$

$$\mathbf{0.394133}$$

A fin de llevar a cabo el análisis experimental de la eficiencia del método, construimos la función ***runningTime*** para la cual dado el tamaño *d* de la instancia

de dimensiones *d x d*, y un numero *n* de instancias podemos calcular el tiempo promedio de ejecución.

```
runningTime[d_, n_] :=
    Module[{instanceSet =
        Table[Table[Table[Random[Integer, {0, 1}], {d}], {d}], {n}]},
        Plus @@ Map[First[Timing[clustering4[#]]] &, instanceSet]/
        Length[instanceSet]]
```

A continuación se muestra el perfil de tiempo de ejecución promedio para tamaños de instancia *n* dentro del rango de 5 a 300 con incrementos de 20. La función es monotónicamente creciente como se esperaba.

```
ListPlot[Table[{d, runningTime[d, 10]},
{d, 5, 300, 20}],
AspectRatio → 0.75, FrameStyle → Thick,
Frame → True, Background → LightBlue, FrameLabel
    → {"Dimension n", "Running Time (sec.)",
"For square matrices of dimension nxn"},
PlotLabel → Running Time of the TSP-based clustering function,
    GridLines → Automatic, Joined → True]
```

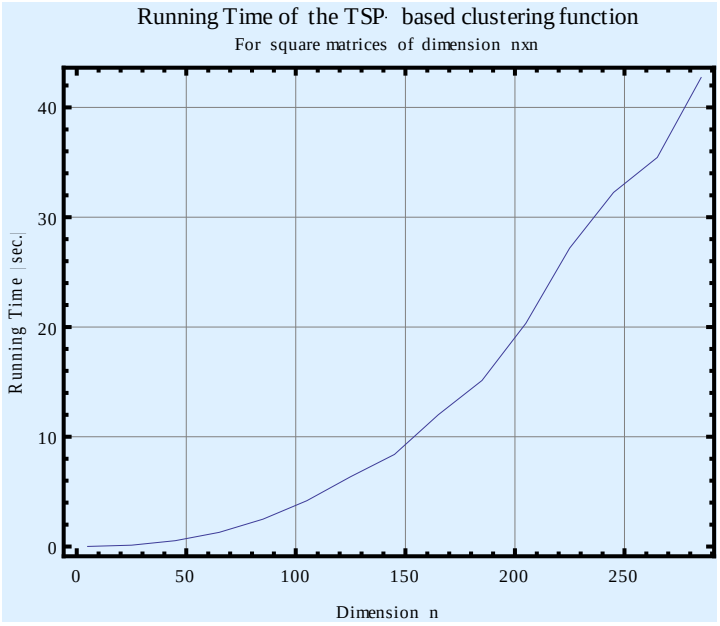


FIGURA 5:17 TIEMPOS DE EJECUCIÓN DEL PROBLEMA TSP BASADO EN CLUSTERING

5.7 Resultados

A lo largo de este capítulo se ha presentado las ventajas de combinar las técnicas de clustering con TSP, para lograr solucionar un agrupamiento de entradas en una matriz binaria. Y con ello solucionar problemas como el que se planteo de encontrar grupos de maestros cuyos intereses afines permitan un trabajo en equipo.

El tiempo de ejecución de las técnicas utilizadas no es realmente significativo, aún cuando la matriz pudiera ser grande y la computadora normal de uso común, por lo que consideramos eficiente el algoritmo.

6 Conclusiones

"Gracias a la Minería de Datos, las computadoras se encargan de seleccionar vastos almacenes de datos. Con una incansable e incesante búsqueda, será posible encontrar la diminuta pepita de oro en una montaña de datos de desperdicio". Edmun De Jesús, octubre de 1995, editor de la revista BYTE Magazine

La contribución principal del proyecto consiste en el diseño de un modelo para analizar información. Los casos de estudio presentados en los capítulos tres y cuatro nos permitieron analizar la información de la base de datos de aspirantes a la Facultad de Informática, se identificaron situaciones importantes como que los aspirantes con mayor puntuación en el examen de admisión para los planes de Ingeniería en Computación e Ingeniería de Software son mujeres, aunque también el porcentaje de mujeres en las carreras antes mencionadas sea menor al 80% del total de los alumnos aceptados. A pesar, de que la base de datos analizada es de poco mas de 5500 registros lo cual la hace considerar pequeña, poca información, se pueden tomar algunos resultados obtenidos para mejorar estrategias de difusión de planes de carrera. Analizar situaciones como aspirantes con resultados con un puntaje mayor a 90 y que aún así no terminan el proceso de ingreso a la Facultad. Desde el punto de vista de aspirantes con calificaciones menores a 50 puntos y que son aceptados, sería interesante analizar su desempeño académico. Los experimentos nos permitieron identificar aquellas escuelas de procedencia cuyos aspirantes han obtenido las mejores resultados oficiales en el examen de admisión.

En el capitulo cinco se presento otra forma de utilizar clustering, en esta ocasión apoyándonos de TSP. Quedó demostrado que esta combinación permite obtener una optimización en la identificación de clusters. Lo cual es relevante desde el punto de vista que nos permite a partir de matrices grandes, ordenar hasta obtener los clusters. En nuestro caso de estudio se refirió a ubicar a los maestros y las materias de interés por grupos de trabajo.

Podemos concluir que la determinación de un proceso correcto de DM, ayuda a identificar cuáles son los registros que no aportan alguna información y que están almacenados innecesariamente, viéndose incrementado el volumen de la información en las bases de datos. De igual forma pueden ser detectados los registros que, no obstante ser considerados a simple vista de poca importancia, son verdaderas fuentes de información que no han sido debidamente explotadas.

A partir de los casos de estudios analizados en el presente trabajo queda la inquietud de analizar también el desempeño académico de los alumnos que ingresaron a la Facultad obteniendo puntuaciones altas en el examen de admisión, así como los tutores asignados, las materias cursadas y los horarios seleccionados, tratando de encontrar información que nos permita hacer mejoras en los planes de carrera, en la distribución de horarios o incluso en la apertura de materias en diferentes estaciones del año u horas. Los análisis mencionados serán motivo de otra investigación que permita contar con bases de datos más complejas.

.

Referencias bibliográficas

[1.] B.Xiao, Q. Zhuge, Y. He, Z. Shao, E. Sha. «Algorithms for Disk Covering Problems with the Most Points.» Texas, USA, 2001.

[2.] Barry, De Ville. Microsoft Data Mining: Integrated Business Intelligence for e-Commerce. Woburn, MA: Digital Press, 2001. Páginas 1-54, 151

[3.] Bezdek, J.C, y S.K. Pal. «Fuzzy Models for Pattern Recognition.» IEEE Press, 1992.

[4.] Boutsidis, C et al. Clustered Subset Selection and its Applications on IT Service Metrics. Napa Valley, CA: ACM Conference on Information and Knowledge Management (CIKM), 2008. Páginas 599-607.

[5.] Bradley, Paul S, Usama M. Fayyad, y Cory A Reina. Scaling EM (Expectation-Maximization) Clustering to Large Data Bases. Redmond Washington: Microsoft Research, 1998.

[6.] Camp, Tracy. «The incredible shrinking.» Communications of the ACM, 1997: 103-110.

[7.] Fayyad U.M., Piatetsky-Shapiro G., Smyth P. Uthurusamy, R. Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, 1996.

[8.] Gan, Guojun, Ma Chaoqun, y Wu Jianhong. Data Clustering: Theory, Algorithms, and Applications. ASA-SIAM, 2007. Páginas 3-65

[9.] Garey, Michael R, Johnson Garey, y S David. Computers and Intractability: A Guide to the Theory of NP-Completeness. NJ: Bell Lab, 1979.

[10.] Garfinkel, R.S. «Motivation and modeling.» En Traveling Salesman Problem,. University of Tennessee, Knoxville: Jonh Wiley & Sons Ltd., 1985. Páginas 17

[11.] Garre Rubio, Miguel, y Mario Charro Cubero. «Estimación del esfuerzo de un proyecto software utilizando el criterio mdl-em y componentes normales n-dimensionales.» Revista de Procesos y Métricas de las Tecnologías de la Información (RPM), 2005.

[12.] Gonzalez, Teofilo F. On the computational complexity of clustering and related problems. Springer Berlin / Heidelberg, 2006.

[13.] Hartigan, John A. Clustering Algorithms. New York: John Wiley & Sons, 1975.

[14.] Holsheimer, A M, y Siebes. Data mining: The search for knowledge in databases. Amsterdam, The Netherlands: In CWI Technical Report CS-R9406, 1994.

[15.] Jain, A K. Data Custering: A Review. ACM, 2000.

[16.] Jain, A.K., M.N.Murty, y P.J. Flynn. «Data Clustering, a review.» ACM Computing Surveys 31, nº 3 (1999). Páginas 265-272

[17.] Jhonson, David S., y Michael R. Garey. Computers and Intractability: A Guide to the Theory of NP-Completeness. Series of Books in the Mathematical Sciences.

[18.] Kaufman Leonard, Rousseeuw Peter J. Finding Groups in Data: An Introduction to Cluster Analysis.

[19.] Kogan Jacob, Nicholas Charles. Grouping Multidimensional Data: Recent Avances in clustering. Berlin Heidelberg: Springer, 2006. Páginas 25, 127

[20.] Kui Shen, Yuan, y David R Karger. U-REST: An Unsupervised Record Extraction SysTem. Alberta, Canada: ACM, 2007.

[21.] Lawler, E.L., J.K. Lenstra, A.H.G. Rinnooy Kan, y D.B. Shmoys. Traveling Salesman Problem.

[22.] Manber, Udi. Introduction to Algorithms - A Creative Approach. MA: Addison-Wesley, 1989.

[23.] Michaels, John G, y Kenneth H Rosen. Applications of Discrete Mathemtics. New York: McGraw-Hill, 1991.

[24.] Power, D.J. A brief history of decision support systems. DSSResources.COM, 2007.

[25.] Salvador, Stan, y Philip Chan. «Determining the Number of Clusters/Segments in Hierarchical Clustering/Segmentation Algorithms.» Dept. of Computer Sciences, Florida Institute of Technology, Melbourne, FL.

[26.] Sharlee, Climer, y Weixiong Zhang. «Take a walk and Cluster Genes: a TSP based aproach to optimal the arragement clustering.» Washington University, 2004.

[27.] Sprague, R.h. and Carlson E. D. Building Effective Decision Support Systems. PrenticeHall, 1982.

[28.] T. Ng, Raymond, y Jiawei Han. «Efficent and Effective Clustering Methods for Spatial Data Mining.» Proceddings of the 20th VLDB Conference, Santiago, Chile, 1994.

[29.] Yu, Chien-Chih. «A Web-based Consumer-Oriented Intelligent Decision Support System for Personalized E-Services.» Sixth International Conference on Electronic Commerce, ICEC´04. 2004.

[30.] Zdravko, Markov, y Daniel T. Larose. Data Minig the WEB, Uncovering Patterns in Web Content, Structure and Usage. New Britain, CT: Wiley, 2007. Páginas 61-112

APENDICE

Apéndice

TABLA 6-1 NORMALIZACIÓN DE CLAVES DE ESCUELAS DE LAS CUALES PROCEDEN LOS ASPIRANTES A LA FACULTAD DE INFORMÁTICA.

CVEESCORI	NOMESCORI	Clave
1	Colegio bachilleres Amealco	1
2	Colegio bachilleres Cadereyta	2
3	Tecnológica agropecuaria cbta (colon)	3
4	Colegio bachilleres Villa Corregidora	4
5	Colegio bachilleres Ezequiel Montes	5
7	Colegio bachilleres El Marques	6
8	Colegio bachilleres Jalpan	7
11	Cbtis 145 San Juan del Rio	8
13	Colegio bachilleres San Juan del Rio	9
14	Incorporada San Juan del Rio	10
15	Particular centro Unión sjr.	11
16	Particular Cambridge Comercio s.j.r.	12
20	Particular Centro Unión Tequis.	13
21	Colegio bachilleres Toliman	14
22	-Conalep Querétaro	15
23	Cetis 105 Santa María	16
24	Cetis 16	17
25	Colegio bachilleres Satélite	18
26	Colegio bachilleres Sta. Rosa Jáuregui	19
27	Colegio bachilleres Azteca	20
29	Cbtis 118	21
30	Prepa Norte UAQ.	22
31	Prepa Sur UAQ.	23
34	Particular San Javier	24
36	Particular Fray Luis de León	25
37	Incorporada Plancarte	26
38	Incorporada 5 de mayo	27
39	Incorporada Alma Muriel	28
40	Particular La paz	29
41	Incorporada Salesiano	30
46	Incorporada Oriente arboledas	31
47	Incorporada Cusva	32

48	Incorporada cudec	33
51	Incorporada Cervantes de Qro.	34
53	Incorporada Gpe Ramírez Álvarez	35
62	Particular Clemencia Borja Taboada	36
63	Particular Cambridge comercio	37
64	Incorporada Iteca	38
67	Particular Estudios Turísticos	39
68	Particular iscca	40
70	Particular Tecnológico Com Vasco de Quiroga	41
72	Incorporada Leonardo de Vinci	42
74	Particular Blas Pascal	43
75	Incorporada Diego Olvera estrada	44
77	Incorporada Colegio Juventud	45
78	Incorporada Clara Barton	46
80	Particular Universidad. Valle de México	47
81	Particular cumdes	48
83	Particular Marcelino Champagnat	49
84	Particular Universidad cudec.	50
85	Particular Liceo	51
86	Colegio bachilleres Tequisquiapán (12).	52
87	Particular Conin	53
88	-Sep video bachillerato	54
90	Particular Anglo mexicano	55
94	Colegio bachilleres pie de la cuesta(13)	56
95	Colegio bachilleres tecnológico Pinal de Amole	57
98	Particular Lafayette de Qro.	58
99	Incorporada Agustín Quiñones m	59
104	Incorporada la providencia	60
105	Colegio bachilleres San Joaquín (14)	61
106	Colegio bachilleres Chichimequillas (15)	62
107	Cecytec Querétaro	63
108	Cecyteq Pedro Escobedo	64
110	Cecyteq Peñamiller	65
112	Cecyteq cerrito colorado	66
114	Incorporada Santiago Galas	67
116	Particular Queretano (sep)	68
128	Particular i.s.s.c.a Ezequiel m.	69
130	Prepa San Juan del rio UAQ.	70

131	Emsad no.2 Jalpan	71
132	Particular Henry Ford	72
133	Particular cnci	73
137	Particular Universidad Marista	74
139	Ceneval	75
312	-Sep abierta	76
314	-Normal del estado	77
348	Particular cudec	78
349	Particular Nuevo Continente	79
402	Baja california norte (pública)	80
408	Chihuahua (pública)	81
411	Guanajuato (pública)	82
413	Hidalgo (pública)	83
417	Morelos (pública)	84
500	" Extranjeros "	85
501	Aguascalientes (privada)	86
511	Guanajuato (privada)	87
517	Morelos (privada)	88
519	Nuevo león (privada)	89
608	Particular la paz	90
620	Prepa semiescolarizada	91
624	Colegio bachilleres (17)	92
625	Particular Universidad Iberomexicana	93