ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

"ESTIMACIÓN DE RIESGO DE FALLO EN MATERIAS BASADO EN SIMILITUD Y MANEJO DE INCERTIDUMBRE DADA POR DATOS ACADÉMICOS HISTÓRICOS"

TESIS DE GRADO

Previa a la obtención del Título de:

INGENIERO EN CIENCIAS COMPUTACIONALES

Presentado por:

ANÍBAL AGUSTÍN VÁSQUEZ CLAD

GUAYAQUIL – ECUADOR

AÑO: 2015

AGRADECIMIENTO

Gracias a mi madre por el amor y el apoyo incondicional, quien me ha acompañado durante toda la vida hasta ahora y nunca sabré agradecerle por todo lo que me ha dado.

DEDICATORIA

Dedicado a mí hermana Mónica, quien me enseñó los valores que me forjaron, la importancia del estudio y a luchar por quienes quiero.

TRIBUNAL DE SUSTENTACIÓN

DR. SIXTO GARCÍA A.

PRESIDENTE

DR. ENRIQUE PELÁEZ J.

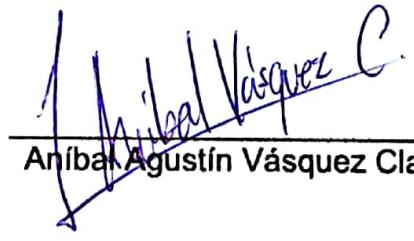
DIRECTOR

DR. XAVIER OCHOA CH.

MIEMBRO PRINCIPAL

DECLARACIÓN EXPRESA

La responsabilidad del contenido de esta Tesis de Grado me corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral.



Aníbal Agustín Vásquez Clad

RESUMEN

La deserción en la educación superior es un problema que no solo afecta a los estudiantes, sino también a las instituciones y universidades, ya que al ofrecer educación incompleta se usan de forma ineficiente los recursos; uno de los factores relacionados a la deserción es la habilidad interpersonal, la cual le permite a un estudiante entablar relaciones con profesores y compañeros, relaciones que contribuyen a culminar de manera exitosa su carrera; el servicio de Consejerías Académicas que ofrece la ESPOL, le permite a los estudiantes entablar una relación con un profesor consejero, quien brinda una guía, semestre a semestre, para que el estudiante pueda mejorar su desempeño, mediante recomendaciones en las materias que debe cursar; las recomendaciones que brinda el profesor consejero son basadas en su experiencia e información académica que visualiza en el sistema de Consejerías, pero este proceso de consejerías podría mejorar su precisión al agregar un factor de estimación de riesgo de reprobación que el estudiante afrontará durante el semestre, dándole un soporte al estudiante sobre la decisión al momento de registrarse en las materias que debe cursar.

El siguiente trabajo propone una arquitectura que utiliza herramientas de clustering y prototipado difuso, para clasificación no-supervisada y predicción a través de la extracción de variables descriptivas. La construcción de prototipos difusos es un método que permite describir a los elementos más

representativos de un cluster, a través de su tipicidad; los prototipos, como los datos más representativos de cada cluster, pueden ser usados en un proceso de clasificación como datos de entrenamiento. Estos prototipos y los clusters pueden ser construidos mediante algoritmos de clustering difuso; además los clusters representados por los prototipos poseen variables descriptivas y atributos que pueden ser asociados a nuevos datos.

Se desarrolló un prototipo de software que implementa el modelo propuesto para estimación de riesgo de reprobación, este software posee tres componentes principales: pre-procesamiento, clustering, y clasificación; dicho software permitió validar el modelo de clasificación para predecir el riesgo de falla en el rendimiento académico de estudiantes, basado en su carga académica y rendimiento, en la selección de cursos antes de registrarse, con un porcentaje de certeza significativo.

Palabras claves: Inteligencia Artificial, Prototipos Difusos, Lógica Difusa, Clasificación No-supervisada, Predicción.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

El servicio de Consejerías Académicas es una herramienta de soporte para la gestión académica de los estudiantes, la cual a través de los profesores asignados como consejeros, permite a los estudiantes tener una recomendación semestre a semestre sobre estrategias que le ayudarían a culminar con éxito su carrera; estas recomendaciones son dadas por la experiencia propia del profesor y son soportadas por información básica sobre el estudiante, tal como: número de materias que ha tomado, número de materias que ha aprobado o reprobado y promedios académicos semestrales; dicha información es usada para que el profesor consejero provea una retroalimentación al estudiante en el proceso de pre-registro en el semestre, etapa en la que el estudiante debe decidir que materias tomará con ayuda del profesor consejero antes de comenzar el semestre; pero esta información no toma en cuenta aspectos como las habilidades a las que un estudiante es afín, el desempeño en cada materia describe de alguna forma qué tan hábil es para resolver problemas de un tipo específico y un estudiante no necesariamente alcanza un desempeño excelente en todas las materias que toma, además las materias que el estudiante tomará en ese semestre implican una carga académica que debe afrontar, puesto que cada materia posee un nivel de dificultad ya sea por el tiempo que debe dedicar o la forma en que es calificada [7]; esta información puede ser incluida como

parámetros en el sistema de Consejerías Académicas, permitiendo incluso predecir o estimar el riesgo de que un estudiante repruebe materias durante el semestre, soportando de una manera más robusta las recomendaciones del profesor consejero.

El siguiente trabajo propone un modelo para estimar el riesgo de fallo en un semestre para un estudiante, donde el fallo implica la reprobación de al menos una materia de las tomadas durante un semestre; dicho modelo está soportado por los datos académicos históricos de todos los estudiantes dentro de la carrera de Ingeniería en Ciencias Computacionales, y carreras equivalentes, los cuales mediante variables que describen el rendimiento y carga académica permiten establecer medidas de semejanza entre estudiantes y medidas de semejanza entre combinaciones de materias, y usando técnicas de minería de datos, estimar el riesgo de fallo para un estudiante; esta estimación apoyaría también al profesor consejero en el proceso de consejerías.

En el capítulo 1 se describe el problema planteado, la justificación de la solución planteada, la propuesta para su resolución y el alcance, los objetivos y la hipótesis de la investigación; en el capítulo 2 se realiza una revisión bibliográfica de los conceptos de clustering, prototipo difuso, variables a extraer y clasificación no supervisada; en el capítulo 3 se analiza y se propone el diseño del prototipo de estimación basado en clustering; en el

capítulo 4 se detalla sobre la implementación del prototipo; en el capítulo 5 se detallan las pruebas realizadas con el prototipo y los experimentos que permitieron definir la configuración del modelo; en el capítulo 6 se discuten las pruebas realizadas y el resultado de los experimentos llevados a cabo, para finalmente describir las respectivas conclusiones, recomendaciones y trabajos futuros.

CAPITULO 1

1. PROBLEMA DE INVESTIGACIÓN

1.1. Antecedentes

Uno de los principales problemas en la educación superior es la deserción académica, esta ocurre cuando un estudiante abandona el programa académico conduciendo a que no complete el programa y en consecuencia no obtenga su título profesional, esto no solo representa algo negativo para los estudiantes, sino también para las universidades e institutos de educación, puesto que se usan de forma ineficiente los recursos al proveer una educación incompleta [68]; usando datos de estudiantes y ex-alumnos obtenidos mediante webservices [62], las estadísticas de crecimiento [66] y el directorio de ex-alumnos de la ESPOL [67], se pudo calcular cifras de deserción académica en la carrera de Ingeniería en Ciencias Computacionales, estas cifras son

mostradas en la figura 1.1; como se puede observar el número de casos de deserción tiende a incrementar mientras que el número de casos de estudiantes graduados ha estado por debajo del número de deserciones en los últimos cinco años; uno de los factores que está ligado a la deserción académica es la habilidad interpersonal, definida como la capacidad para comunicarse de manera exitosa con otras personas [69], según McGaha y Fitzpatrick este factor está relacionado a la deserción debido a que las relaciones entre un estudiante, sus compañeros y profesores contribuyen a la realización exitosa de su carrera; el servicio de Consejerías Académicas permite a los estudiantes entablar una relación con los profesores, quienes actúan como consejeros académicos con el objetivo de guiar a los estudiantes para que mejoren su desempeño [70] mediante recomendaciones en las materias que debe cursar o está cursando; el profesor consejero usa su experiencia para determinar qué tan difícil podría ser para un estudiante aprobar una materia que debe cursar, y de acuerdo a sus datos académicos el profesor sugiere cuantas materias debería tomar considerando la dificultad que le representa al estudiante, según su criterio, aprobar cada una de ellas; este proceso podría ser soportado por una predicción del rendimiento

que el estudiante va a tener al finalizar el término académico permitiendo al profesor dar una recomendación más precisa.

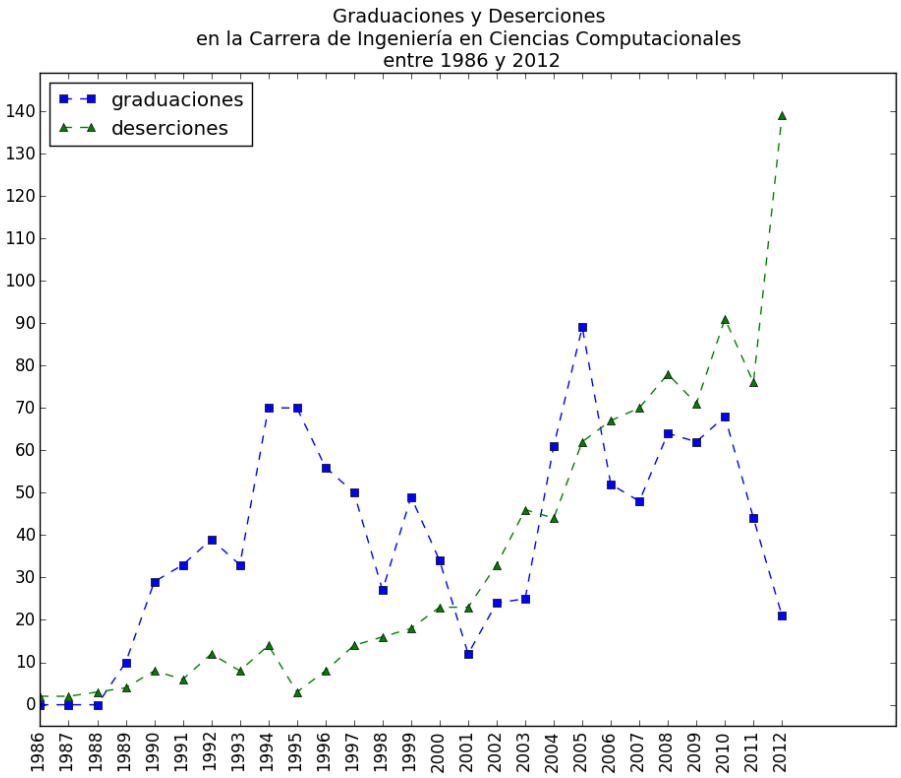


Figura 1.1. Graduaciones y deserciones durante 1986 y 2012 en la carrera de Ingeniería en Ciencias Computacionales

La predicción del rendimiento académico en estudiantes universitarios ha sido objeto de estudio en años recientes, ya que el monitoreo y evaluación del desempeño académico es fundamental en el sistema educativo; muchas metodologías desarrolladas usan técnicas de minería de datos e inteligencia artificial para así dar un pronóstico a los estudiantes antes de tomar un curso o registrarse en un semestre; esta predicción de

rendimiento generalmente posee una precisión mayor al 70% usando estas técnicas, las cuales se soportan en datos que reflejen el desempeño académico, información de nivel socioeconómico y demográfico [63], [64] y [65].

Una técnica de minería de datos comúnmente utilizada para la construcción de un modelo de predicción, es el análisis probabilístico soportado por datos históricos [3]; por ejemplo, la estimación de riesgo accidental, que utiliza modelos basados en datos históricos de las aseguradoras [4]; dentro de estas técnicas de minería de datos se encuentra el clustering, la cual es una técnica utilizada en el aprendizaje no supervisado de máquinas, esta técnica permite clasificar, categorizar o agrupar datos, dependiendo del contexto en que se utilice, de acuerdo a medidas de similitud entre datos; generalmente esta técnica se realiza mediante algoritmos de aprendizaje no supervisado y uno de los objetivos principales de estos algoritmos es la identificación de clusters o agrupaciones de datos, esto para la posterior clasificación de nuevos datos. Una de las características de los clusters formados es la homogeneidad que presentan con respecto a los datos que agrupan y su distribución de atributos, en relación a los datos de otros de clusters [1].

Para la minería de datos, el promedio de un estudiante podría ser uno de los elementos que describen el desempeño; sin embargo esta medida no considera de forma concreta otros elementos que influyen directamente el rendimiento de un estudiante ni la carga académica a la que se somete en un semestre [5]. El rendimiento de un estudiante puede ser expresado de forma relativa, ya que un estudiante puede tener un buen rendimiento en materias que posean un gran elemento práctico y no tenerlo en materias con un fuerte elemento teórico, es decir que el rendimiento de un estudiante no necesariamente puede ser tomado como una medida generalizada y única, ya que también se deben considerar las habilidades que un estudiante adquiere luego de aprobar con éxito una materia, según esto se podría descomponer el rendimiento en medidas relativas a grupos de materias interrelacionadas que requieren el desarrollo de una habilidad específica para ser aprobadas [7]; También es posible que este rendimiento dependa de la carga académica a la que es sometido un estudiante en un semestre, dada por la dificultad de las materias que toma y la rigurosidad con la que estas son calificadas [6], es decir que una combinación de materias tomadas en un semestre poseen un nivel de dificultad y rigurosidad que afectan el rendimiento de un estudiante. Estas medidas puedan ser usadas para alimentar un

proceso de clustering que permita clasificar estudiantes y estimar el riesgo de reprobación de nuevos estudiantes que comparten características con los ya clasificados.

1.2. Descripción del problema

Un estudiante al tomar un conjunto de materias en un semestre se enfrenta al riesgo de fallar en alguna de estas materias, el fallo en estas materias puede ocurrir por muchas razones, algunas de estas están relacionadas con factores emocionales o de comportamiento, que según McGaha y Fitzpatrick [60] son difíciles de medir o modelar matemáticamente y aunque existen algunos métodos de abstracción del comportamiento, como la contingencia de características personales y características sociales , estos factores también son difíciles de medir, además de que requieren un alto costo para ser recolectados y tienden a envejecer, por lo que deben ser actualizados continuamente; también se debe considerar que la mayoría de las variables categóricas recolectadas no necesariamente describen una circunstancia real, de la misma manera el categorizar un estudiante, basado solo en su rendimiento general, no necesariamente implica que pertenece a una sola categoría de estudiante ya que es posible que cumpla ciertas características, dadas por otras variables relacionadas a su

rendimiento, que indiquen que pertenece a más de una a la vez [1]; estas categorías de estudiantes poseen ciertas variables descriptivas, como la tasa de reprobación, la cual puede ser generalizada a todos los estudiantes dentro de una categoría, y puede ser usada para estimar el riesgo de falla en una materia.

También se debe considerar que el riesgo de falla en una materia podría implicar una relación con la combinación de materias tomadas durante el semestre, esta relación se puede definir como carga académica semestral, y es posible que mientras mayor sea la carga académica para un estudiante, mayor sea el riesgo de fallo en una materia para esa combinación de materias en el semestre, pero la carga académica no solamente se encuentra en función de cuantas materias se ha tomado, sino más bien de las materias en sí, pues cada materia tiene asociada un nivel de dificultad, debido al material cubierto o a la vinculación con conceptos abstractos, así mismo de acuerdo a que tan rigurosa es evaluada por el profesor; estas medidas son difíciles de obtener, ya que requieren un estudio exhaustivo de información subjetiva por parte de estudiantes y profesores, además la información recolectada es difícil de actualizar y mantener para poder asegurar futuras investigaciones o predicciones basadas en los datos a largo plazo.

Las calificaciones de los estudiantes y el registro de aprobación son medidas que están siempre disponibles y son actualizadas y recolectadas de forma periódica, pero estas calificaciones no proveen mayor información que un historial de actividad académica, aunque de alguna forma pueden aportar a describir la carga académica semestral y el rendimiento de un estudiante, requieren un procesamiento y soporte de datos históricos para poder ser usadas en el contexto requerido. Las calificaciones de los estudiantes al ser recolectadas periódicamente, su acceso no es costoso y al permitir redefinirlas en el contexto del rendimiento por estudiante y carga académica, se podrían usar en un proceso de categorización de estudiantes, mediante técnicas de clasificación no-supervisada, y así estimar, para nuevos estudiantes, el riesgo de fallo al que se enfrentarían al tomar una combinación de materias durante un semestre.

1.3. Justificación

La institución ha estado impulsando una herramienta de soporte a los estudiantes mediante el servicio de Consejerías Académicas, a través de la cual los profesores pueden consultar el avance en la carrera y el cumplimiento de requisitos en el proceso formativo de un estudiante al que es asignado como consejero, para así poder

recomendar según su experiencia como profesor una estrategia para culminar con éxito el semestre; estas recomendaciones hechas por el profesor son soportadas por información básica relacionada con el número de materias tomadas, por tomar, reprobadas y el promedio académico; por lo tanto, es necesario dotar al profesor consejero de otro tipo de información relacionada con el potencial riesgo de fallo en alguna materia y sus posibles recomendaciones hacia el estudiante, además esta información debería de estar disponible en tiempo real; la información mostrada como indicadores, y basada en el modelo de estimación propuesto en este proyecto, puede ayudar al consejero académico en su recomendación respecto a la dificultad y rigurosidad de las materias que un estudiante desea tomar o esté cursando en un semestre considerando el perfil del estudiante. Este soporte a las recomendaciones del consejero académico, como la dificultad, rigurosidad asociados a una materia y riesgo de reprobación en un semestre, podrían influir en la disminución del porcentaje en estudiantes que alargan su estancia en la universidad, se retiran o pierden la carrera debido a que no poseen información oportuna ni herramientas de soporte en el proceso de decidir que materias tomar en un semestre.

1.4. Propuesta y alcance

El principal resultado esperado en esta investigación es definir un modelo de estimación de riesgo de reprobación de materias, que un estudiante va a tomar en un semestre, en base a su historia académica, soportado por datos históricos y su comparación con otros estudiantes con historia académica similar. Se espera definir la similitud de la historia académica de un estudiante basado en el número de materias que aprueba, número de materias que reprueba, calificaciones que obtiene y la dificultad que representan las materias por semestre, incluyendo su rigurosidad en cómo se califican y la distribución de notas que presenta. Finalmente se espera construir un prototipo que puede ser incorporado al sistema de Consejerías Académicas como un semáforo que le permita al consejero académico proveer de información oportuna al estudiante antes de registrarse en un nuevo, presentada como una escala de riesgo. Esta investigación incluye también:

1. Desarrollar un modelo de estimación de riesgo de fallo en materias basado en técnicas de clustering y clasificación no supervisada, que será implementado en el prototipo.

2. Desarrollar medidas relacionadas a las habilidades adquiridas por un estudiante, en función de las materias que aprueba exitosamente.

1.5. Objetivos

1.5.1. Objetivo general

El objetivo general de este trabajo es proponer un modelo de manejo de incertidumbre para estimación de riesgo en el fallo de materias de un estudiante considerando un conjunto de materias que tomará en un semestre, soportado por la historia académica del estudiante y los datos académicos de estudiantes similares, a partir del rendimiento y la carga académica semestral.

1.5.2. Objetivos específicos

- Definir las posibles variables que influyen en el desempeño académico de un estudiante dado por el rendimiento previo del estudiante y la carga académica en un semestre.

- Determinar que variables son más idóneas para categorizar a los estudiantes de Ingeniería en Ciencias Computacionales.

- Determinar la similitud entre estudiantes basada en su historia académica.

- Diseñar e implementar un prototipo que permita estimar el riesgo para un estudiante de acuerdo a sus características, datos históricos y comportamiento similar con otros estudiantes.

1.6. Pregunta de investigación e hipótesis

Cada materia parte de la malla curricular aporta a la formación de un estudiante cuando este la aprueba con éxito o no, estas materias está asociada con un área del conocimiento o habilidad que un estudiante debe desarrollar, por ejemplo existen materias relacionadas con la programación, mientras otras materias están relacionadas con cálculo o física, dado que las calificaciones de un estudiante reflejan en general el desempeño en una materia **¿es posible determinar la similitud de un estudiante con otros, basado en el desempeño obtenido en materias similares y relacionadas?** Siguiendo con el ejemplo, un estudiante que tiene un buen desempeño en materias relacionadas con la programación nos podría indicar que posee el requisito para aprobar materias que demandan esta habilidad, así mismo si otro estudiante posee un desempeño similar, se podría intuir que los dos son semejantes, de alguna manera, con respecto a otros que no adquirieron esa

habilidad; también, podríamos decir que un estudiante que toma materias de nivel básico, relacionadas con la programación, y otro estudiante que toma materias de un nivel avanzado, también relacionadas con la programación, requerirán desarrollar la misma habilidad para programar, es decir que independientemente del nivel en que esté un estudiante el desempeño en materias relacionadas puede ser medido, al igual que el nivel de desarrollo de la habilidad requerida para aprobarlas, pero **¿dichas medidas permitirían definir la similitud entre varios estudiantes?**

Al definir la similitud entre estudiantes de diferentes niveles podríamos decir que, para un estudiante de menor nivel, que comparte características similares con otro estudiante de mayor nivel, el desempeño durante la carrera podría ser similar para ambos; esto es, si el estudiante de menor nivel se enfrenta a las mismas dificultades que el estudiante de mayor nivel, es muy probable que tenga el mismo riesgo de reprobarlas; la hipótesis es que, conociendo estas similitudes se podría estimar el riesgo de reprobar una materia durante el semestre que un estudiante afronta antes de tomar dichas materias, de manera que pueda reevaluar la carga académica y disminuir el riesgo de reprobación.

1.7. Metodología

Como primer paso se realizará una revisión literaria de las diferentes alternativas de algoritmos de clustering, clasificación no supervisada y los diferentes índices de validación en los métodos de clustering, también se revisará metodologías utilizadas para la construcción de prototipos difusos basados en algoritmos de clustering y las diferentes medidas de disimilitud y semejanza.

Los datos académicos históricos que serán usados se obtendrán mediante el consumo de webservices de calificaciones que provee el Centro de Servicios de Información (CSI) de la Institución [62].

Después de analizar las diferentes alternativas de las metodologías, utilizadas en clustering y construcción de prototipos, se realizará un levantamiento de los requerimientos funcionales y no funcionales del software prototipo a implementar, cuyo diseño será basado en componentes y con una arquitectura orientada a servicios; también se especificarán las interfaces diseñadas para acceder a esta funcionalidad, que podrá ser implantada dentro del sistema de Consejerías Académicas.

Luego se procederá a diseñar los componentes principales del sistema los cuales son: Componente de Pre-Procesamiento, el cual transformará, los datos obtenidos a través del webservice, a

variables difusas y relacionados con estudiante, semestre y materia; el Componente de Clustering el cual realiza el proceso para crear el cluster multinivel en la arquitectura propuesta, y el Componente de Clasificación el cual tendrá como entrada los datos académicos del estudiante y las materias que desea tomar en un semestre, y obtendrá los índices asociados al riesgo de reprobar las materias que desea tomar; este último componente también será el encargado de clasificar los datos de entrada en el clustering multinivel de acuerdo a la data previamente ingresada. El módulo será sometido a pruebas de acuerdo al estándar IEEE 829-2008 para pruebas documentadas de software, así mismo se realizarán pruebas de integración con los componentes antes mencionados y las pruebas de rendimiento.

A continuación se elaborarán pruebas que permitan analizar el nivel de precisión de las estimaciones para lo cual se diseñará un experimento, usando los datos de los estudiantes de Ingeniería en Ciencias Computacionales durante los términos académicos entre 1978 y 2012 para el proceso de clustering, lo cual emulará la estimación para estudiantes que tomaron materias en los términos I y II del 2013; y realizar una comparación con datos reales dando como producto un análisis descriptivo de la precisión del modelo. Finalmente se analizarán los resultados para generar las

conclusiones y recomendaciones pertinentes para futuras investigaciones o implementaciones.

CAPITULO 2

2. REVISIÓN BIBLIOGRÁFICA

2.1. Algoritmos de clustering

Los algoritmos de clustering permiten realizar sub-agrupaciones para un conjunto de observaciones X dado [1], estas sub-agrupaciones pueden ser dadas en un contexto de estructuras internas dentro del conjunto X descritas por una similitud entre los elementos pertenecientes a X; estos algoritmos son una base fundamental para el agrupamiento y clasificación de datos no supervisado [8]. Los algoritmos antes mencionados consisten de una serie de procesos que permiten determinar las agrupaciones o clusters dentro de un data set o conjunto de datos; estos procesos son soportados por medidas de similitud, distancias o medidas de

afinidad que permiten identificar las relaciones entre datos $x \in X$ para su agrupamiento, clasificación, categorización o clusterizado. De forma general el resultado de alguno de estos algoritmos es un subconjunto de datos o muestras del data set original, con la característica de que cada punto dentro de este subconjunto o muestra poseen atributos en común; los clusters producto de este proceso se denotan formalmente como:

$$C_r = \{y_1, y_2, \dots, y_m\} \quad y_i \in X, i = 1,2, \dots, m \tag{2.1}$$

$$X = \bigcup_{r=1}^{c} C_r \tag{2.2}$$

Donde:

C_r es un clúster tal que $C_r \subseteq X$

X es el data set tal que $X \subseteq \mathbb{R}^p$

2.1.1. K-Means

K-means es un algoritmo de clustering basado en la generalización de la media sobre muestras ordinarias para grandes conjuntos de datos de p dimensiones. Este es un método de clustering que permite particionar un conjunto X en k subconjuntos de p dimensiones, estos subconjuntos mantienen ciertas características de consistencia de acuerdo a una variación interna de clases, esta

variación interna de clases está definida por: la distancia de cada punto a la media más cercana y la distancia entre las medias [10]; la variación interna de clases es independiente en cuanto al orden en que los datos son procesados [8] [9]. Actualmente existen varias implementaciones de este algoritmo y debido a su bajo costo computacional tiene un uso muy común para agrupación de grandes conjuntos de datos, predicción no lineal, aproximación de distribuciones multivariadas, descripción de comportamiento no paramétrico y clasificación. Para su utilización es requerido que todas las variables sean cuantificables y que además exista una interpretación de disimilitud para la distancia Euclidiana $d(x_i, x_{i'})$ definida como:

$$d(x_i, x_{i'}) = \sum_{j=1}^{p} \left(x_{ij} - x_{i'j}\right)^2 = \|x_i - x_{i'}\|^2 \tag{2.3}$$

K-means se basa en la minimización de $W(C)$; mediante la asignación de n puntos del data set al clúster K, tal que la disimilitud promedio de los puntos que pertenecen al clúster K y su media sea minimizada, este criterio a minimizar es definido como:

$$W(C) = \sum_{k=1}^{K} N_k \sum_{x_i \in C} \|x_i - \bar{x}_k\|^2 \tag{2.4}$$

Donde:

$$N_k = \sum_{i=1}^{n} I(C(i) = k) \tag{2.5}$$

$$\bar{x}_k = \left(\bar{x}_{1_k}, \bar{x}_{2_k}, \dots, \bar{x}_{p_k}\right) \tag{2.6}$$

$$C(i) = \arg\min_{1 \le k \le K} \|x_i - \bar{x}_k\|^2 \tag{2.7}$$

Este método de clustering generalmente usa la validación mediante el índice de Dunn [61], este índice es una métrica para identificar si el proceso de clustering dio como resultado clusters compactos y a la vez bien separados; el índice de Dunn se define como:

$$DI_m = \min_{1 \le i \le K} \left\{ \min_{1 \le j \le K, j \ne i} \left\{ \frac{\delta(C_i, C_j)}{\max_{1 \le r \le K} \Delta_r} \right\} \right\} \tag{2.8}$$

Donde:

$\delta(C_i, C_j)$ es la distancia entre clusters

$\Delta_r = \max_{x,y \in C_r} \{d(x, y)\}$ es la máxima distancia dentro del clúster r.

2.1.2. Propagación por afinidad

Es un método de clustering que toma como entrada la definición de las medidas de similitud entre dos puntos y como resultado se producen “ejemplares” encontrados de manera aleatoria, este es un proceso iterativo y se da por el intercambio de mensajes entre datos hasta que emerja un conjunto de alta calidad de los llamados ejemplares y sus respectivos clusters [11]. Los ejemplares pueden ser establecidos mediante distancias euclidianas entre puntos, para esto primero se toma a cada punto como posible ejemplar, luego se transmite de forma recursiva valores dentro del grafo que representa el data set, donde cada punto del data set es un nodo del grafo y los mensajes son transmitidos por las aristas del grafo, la magnitud de cada mensaje describe la afinidad de un punto a otro para elegir a otro punto como su ejemplar.

En este método se proporciona como entrada un conjunto de valores que describen la afinidad entre datos, definiendo $s(i,k)$ como la afinidad que tiene el punto k-ésimo para describirse como el ejemplar del punto i-ésimo; una forma de especificar esta medida es con la distancia Euclidiana entre el punto i y el punto k. El procedimiento también requiere especificar otros datos de entrada como el número de clusters, y una medida de afinidad

$s(k,k)$ que representa la afinidad de un dato a sí mismo, este valor es usado en el criterio del "mejor valor" $s(k,k)$ para elegir el ejemplar, el "mejor valor" $s(k,k)$ es llamado "preferencia". El número de ejemplares identificados es equivalente al número de clusters que se obtendrán como resultado y son influenciados por las preferencias de entrada; además, estos son consecuencia del proceso de paso de mensajes. En casos particulares en que los valores de preferencia de entrada tienen valores distintos, la preferencia resultante tiende a ser un valor común para cada ejemplar y generalmente es la media aritmética de los valores de similitud de entrada. Existen dos tipos de mensajes que pueden ser pasados entre puntos: un mensaje de "responsabilidad" y un mensaje de "disponibilidad".

El mensaje de responsabilidad definido como $r(i,k)$ es enviado de un punto i a un candidato a ejemplar k y define que tanto describe el punto k al punto i como un ejemplar, tomando en consideración los potenciales ejemplares para i; el mensaje de responsabilidad se denota como:

$$r(i,k) = s(i,k) - \max_{k' \neq k}\{a(i.k') + s(i,k')\} \qquad (2.9)$$

Donde:

$s(i,k)$ describe la afinidad del punto k al punto i

$a(i,k)$ es el mensaje de disponibilidad

El mensaje de disponibilidad $a(i,k)$ describe que tan apropiado es para el punto i elegir el punto k como su ejemplar; para cada punto k el valor de disponibilidad es inicialmente cero y los mensajes de responsabilidad son calculados usando la fórmula anteriormente definida.

En la primera iteración $r(i,k)$ es inicializado con el valor de afinidad entre i y k; luego en cada iteración el valor de disponibilidad es definido de la siguiente forma:

$$a(i,k) = \min\left\{0, r(k,k) + \sum_{i' \notin \{i,k\}} \max\{0, r(i',k)\}\right\} \qquad (2.10)$$

Para limitar la posibilidad de grandes valores de responsabilidad de un punto hacia sí mismo la definida “auto-disponibilidad” está dada por:

$$a(k,k) = \sum_{i' \neq k} \max\{0, r(i',k)\} \qquad (2.11)$$

El paso de mensajes solo se da entre pares de puntos donde el valor de afinidad es conocido, y la iteración se realiza hasta: lograr

la optimización en $\max_{i,k}\{a(i,k)+r(i,k)\}$, identificando a i como un ejemplar; para $i=k$ se debe lograr un número determinado de iteraciones o alcanzar un umbral para $a(i,k)$ y $r(i,k)$.

2.1.3. Clustering aglomerativo

El clustering aglomerativo es un paradigma del clustering jerárquico, el cual es un método que permite construir clusters de manera jerárquica, es decir, agrupando datos mediante particiones con características cada vez más generales; el clustering jerárquico difiere de las otras técnicas, tal como k-means, al no tener como valor de entrada el número de clusters que se darán como producto del proceso, y se debe proveer además una medida de disimilitud entre grupos, basado en observaciones [8] [13]. Existen 2 paradigmas básicos para el clustering jerárquico: aglomerativo (bottom-up) y divisivo (top-down). El paradigma aglomerativo es el de empezar con datos indivisibles y agruparlos de forma recursiva hasta tener clusters que ya no puedan ser fusionados, el criterio de fusión se da en función de la menor disimilitud entre clusters. El paradigma divisivo empieza con todo el conjunto y este es particionado de forma recursiva, cada partición de un clúster se da en función de la mayor medida de disimilitud entre clusters. Al ser métodos de clustering jerárquico los clusters

resultados están asociados unos a otros a través de la medida de disimilitud que permiten rastrear el origen de cada uno de los datos en el orden de los niveles con los que nuevos clusters son generados. Los métodos aglomerativos poseen una propiedad de monotonicidad, es decir la disimilitud entre clusters unidos es monótonamente creciente con relación al nivel de fusión; si el clustering jerárquico es visto como un árbol producto de cada proceso de fusión recursiva, la altura de cada nodo es proporcional al valor de disimilitud entre clusters de sus hojas.

En cada uno de los $N-1$ pasos del clustering aglomerativo los clusters más cercanos son fusionados en el mismo clúster dando como resultado un clúster menos en el siguiente nivel, para la fusión se define una medida de disimilitud entre clusters, siendo G y H la representación de dos grupos, la disimilitud $d(G,H)$ entre G y H es calculada del conjunto de pares ordenados en el data set con una disimilitud d_{ij} donde un miembro del par ordenado i pertenece a G y el otro miembro j pertenece a H. En el Clustering Aglomerativo de Vínculo Simple (Single Linkage SL) se toma la disimilitud entre clusters como la menor medida de disimilitud en un par ij, tal que:

$$d_{SL}(G,H) = \min_{i \in G \wedge j \in H} \{d_{ij}\} \quad (2.12)$$

Esta técnica también es llamada del vecino cercano (nearest-neighbor technique). El Clustering Aglomerativo de Vínculo Completo (Complete Linkage CL) toma la disimilitud intergrupal como la máxima medida de disimilitud entre un par ij, también conocida como la técnica del vecino lejano (furthest-neighbor technique), definida como:

$$d_{CL}(G,H) = \max_{i \in G \wedge j \in H} \{d_{ij}\} \quad (2.13)$$

El Clustering Aglomerativo del Promedio de Grupo (Group Average GA) es una técnica que utiliza el promedio de la disimilitud sobre los datos que pertenecen a los grupos G y H, y es definida como:

$$d_{GA}(G,H) = \frac{1}{N_G \cdot N_H} \sum_{i \in G} \sum_{j \in H} d_{ij} \quad (2.14)$$

Donde N_G y N_H son las respectivas cardinalidades del grupo G y H

Los clusters producto de SL pueden violar la propiedad de compacidad, esta propiedad de compacidad expresa que los clusters creados deben ser compactos son internamente y disímiles son con respecto a otros clusters, por lo cual todos los clusters tienden a ser similares a otros basados en las disimilitudes

provistas, si definimos el diámetro de un grupo G (DG) como la mayor medida de disimilitud entre los miembros de su grupo, formalmente como:

$$DG = \max_{i,i' \in G} \{d_{ii'}\} \qquad (2.15)$$

SL hace que los clusters tengan una tendencia a tener un diámetro muy largo, es decir que la propiedad de compacidad de los clusters es violada. La técnica CL representa un extremo opuesto a SL, los grupos G y H son considerados cercanos si todos los datos en la unión son relativamente similares, esto tiende a crear clusters más compactos y con un diámetro menor produciendo clusters que violan la propiedad de cercanía ya que los datos asociados a un clúster pueden ser mucho más similares a miembros de otros clusters que a miembros de su propio clúster.

El GA es un enfoque intermedio al de SL y CL, tiende a producir clusters relativamente cercanos entre sus miembros y relativamente lejanos unos de otros, este resultado depende de la escala en que se da la medida de disimilitud d_{ij}, dado que GA tiene una propiedad de consistencia estadística, en cuanto a las propiedades violadas por SL y CL, se puede argumentar que la influencia de la invarianza de la medida de disimilitud d_{ij} no siempre expone una mayor complicación; asumiendo que para

cada par de atributo-valor en el data set $X_T = (X_1, X_2, \dots, X_p)$, cada clúster k es una muestra aleatoria con una densidad poblacional $p_k(x)$, entonces el data set completo es una muestra aleatoria de la mezcla de K densidades poblacionales, así la medida de disimilitud del GA Clustering $d_{GA}(G, H)$ está estimada por [8]:

$$\int \int d\,(x, y) p_G(x) p_H(y) dx dy \tag{2.16}$$

Donde $d(x, y)$ es la medida de disimilitud entre los puntos x y y en el espacio de los pares atributo-valor.

2.1.4. Fuzzy C-Means

Fuzzy C-Means (FCM) es un algoritmo de clustering, que permite representar la partición Y_r de un conjunto X, como una matriz $U_{c\times N} = [u_{ik}]$ donde cada uno de estos valores $u_{ir} \in [0,1]$ representan los valores de membresía de un elemento x_i a cada partición Y_r; en el contexto de lógica difusa cada uno de estos valores son tomados como el valor de membrecía de un punto a uno de los clusters y a su vez estos clusters son considerados conjuntos difusos [2] [14]. FCM hace uso del criterio más popular para identificación óptima de particiones difusas, el cual está asociado con la generalización de least-squared errors, esto es:

$$J_m(U, w, c) = \sum_{i=1}^{n} \sum_{r=1}^{c} (u_{ir})^m \, \|x_i - w_r\|_A^2 \tag{2.17}$$

Donde:

$X = x_1, x_2, \ldots, x_n \subseteq R^p$ es el data set

c es el número de clusters en X y cumple con $2 \leq c < n - 1$

m el exponente de ponderación

U es la matriz de partición difusa de Y

$w = \{w_1, w_2, \ldots, w_c\}$ representa los centroides estimados

$w_i = \left(w_{i_1}, w_{i_2}, \ldots, w_{i_n}\right)$ representa el centroide del clúster i

$\|\|_A$ la norma-A en R^p

A es la definición de la matriz de peso $(n \times n)$

La distancia cuadrada entre y_k y v_i se calcula por la norma-A como:

$$d_{ir}^2 = \|x_i - w_r\|_A^2 = (x_i - w_r)^T A (x_i - w_r) \tag{2.18}$$

Según esta definición se agrega un peso a cada error cuadrático, este peso es igual a $(u_{ik})^m$, es decir la membresía de x_i en el clúster i elevado al exponente de ponderación. Los centroides son

una representación del centro de masa del subconjunto particionado. Si $m = 1$ se puede observar que J_m solo minimiza la matriz U de tal forma que los valores de membresía son 1 o 0 y los correspondientes v_i son centroides geométricos de las particiones C_r, asumiendo esto podemos descomponer J_m en [2]:

d_{ir}^2 distancia cuadrada desde el punto x_i al centroide w_r

$d_{ir}^2 (u_{ir})^m$ error al cuadrado dada por los valores de membresía de x_i en el clúster i como los pesos

$\sum_{r=1}^{c} d_{ir}^2 \, (u_{ir})^m$ sumatoria de los errores al cuadrado a través de los $x_i \in C_r$ reemplazados parcialmente por todos los centroides c en w_r

$\sum_{i=1}^{N} \sum_{r=1}^{c} d_{ir}^2 \, (u_{ir})^m$ sumatoria ponderada sobre todos los puntos hacia de X hacia los centroides w.

El exponente de ponderación m controla los pesos relativos en cada uno de los errores cuadrados d_{ir}^2, si incrementamos m existe una tendencia de degradación de los valores de membresía, la determinación de m depende en gran medida de los datos, por lo m se determina de forma empírica; aunque para data sets extensos $1.5 \leq m \leq 3.0$ da buenos resultados, y en muchas investigaciones es común un $m = 2.0$ [39].

La norma-A también tiene una gran influencia en el resultado [39], esta determina la distancia relativa de los datos a sus centroides así como la distancia de un centroide a otro centroide, entre las norma-A se encuentran la distancia euclidiana, norma diagonal, norma de Mahalanobis y el producto interno sobre R^n. Para distancias euclidianas los clusters son identificados como hiper-esferas, mientras que para distancias no Euclidianas los clusters son identificados como hiper-elipses. Para cada clúster difusos de X se define un par ordenado (U, w) tal que minimiza localmente J_m, si $x_i \neq w_r$ para cada i y r el par ordenado (U, w) optimiza localmente a J_m sólo si:

$$w_r = \sum_{i=1}^{n} (u_{ir})^m \,; 1 \leq r \leq c \tag{2.19}$$

$$u_{ir} = \left(\sum_{s=1}^{c} \left(\frac{d_{ir}}{d_{is}} \right)^{2/(m-1)} \right)^{-1} \,; 1 \leq k \leq N; 1 \leq i \leq c \tag{2.20}$$

Donde:

$d_{ir} = \|x_i - w_r\|_A$ define la distancia mediante la norma-A entre x_i y el centroide w_r en el clúster r

$d_{is} = \|x_i - w_s\|_A$ define la distancia mediante la norma-A entre x_i y el centroide w_s en el clúster s

Estas expresiones son necesarias para proveer significado a la optimización de J_m mediante una iteración simple de Picard, es decir hasta que la iteración muestre pocos cambios en U y w.

Algoritmo Fuzzy c-Means (FCM):

1. Inicializar c, m, A, ε y $\|\|_A$. Elegir una matriz inicial $U^{(0)} \in M_{fc}$. Para $r = 0,1,\ldots,LMAX$

2. Computar $w_r, r = 1,2,\ldots,c$ con la ecuación anterior

3. Computar y actualizar la matriz de membresía $U^{(r+1)} = \left[u_{ir}^{(r+1)}\right]$ con la ecuación anterior

4. Comparar $U^{(r+1)}$ con $U^{(r)}$ con una norma de matriz definida. Si $\left|U^{(r+1)} - U^{(r)}\right| < \varepsilon$, el algoritmo se detiene. Caso contrario asignar $U^{(r+1)} = U^{(r)}$ y retornar al paso 2

2.1.5. Possibilistic C-Means

Esta es una variación de FCM que usa una función de membresía posibilista que describe el grado de pertenencia en un elemento hacia a un clúster. Para datos representativos hay un valor de membresía alto, mientras que para datos no representativos el

valor de membresía es bajo [15]. La función objetivo es definida de la siguiente forma:

$$\min\left\{J_m(U,v,c) = \sum_{i=1}^{n}\sum_{r=1}^{c} d_{ir}^2\,(u_{ir})^m + \sum_{i=1}^{n}\eta_r \sum_{r=1}^{c}(1-u_{ik})^m\right\} \qquad (2.21)$$

Donde:

$X = x_1, x_2, \ldots, x_n \subseteq R^p$ es el data set

d_{ir}^2 distancia cuadrada desde el punto x_i al centroide w_r

u_{ir} es el valor de membresía del punto x_i en el clúster r

m el exponente de ponderación

η_r es un número positivo definido para cada clúster r

c es el número de clusters en X y cumple con $2 \leq c < n - 1$

n es la cardinalidad de X

El valor de membresía u_{ir} puede ser definido como:

$$u_{ir} = \frac{1}{1 + \left(\frac{d_{ir}^2}{\eta_r}\right)^{\frac{1}{m-1}}} \qquad (2.22)$$

El valor de η_r determina la distancia relativa relacionada para el valor de membresía, este es definido como:

$$\eta_r = \frac{\sum_{i=1}^{n} (u_{ir})^m \, d_{ir}^2}{\sum_{i=1}^{n} (u_{ir})^m} \qquad (2.23)$$

2.2. Prototipado difuso

El objetivo del prototipado es construir un elemento que pueda representar de forma generalizada los datos pertenecientes a un grupo, esto se basa en que existe una noción de tipicidad que permite establecer diferencias y caracterización entre los datos de un grupo dado; la tipicidad para un elemento específico que pertenece a una categoría, está dado por la semejanza a los miembros de la misma categoría y la disimilitud con respecto a los miembros de otras categorías [16] [17] [18].

Para poder definir la semejanza y disimilitud de datos se debe establecer la definición de las medidas de comparación en un contexto matemático; si asumimos que los datos pueden ser representados como datos espaciales por ejemplo se pueden usar medidas de comparación, como medidas que dependan de distancias entre los datos, que permitan definir la semejanza y disimilitud; esta función dependiente de la distancia entre los datos usualmente es un valor que se encuentra en el intervalo de $[0.0,1.0]$.

2.2.1. Medidas de disimilitud

2.2.1.1. Medidas de distancia

La medida de disimilitud es generalmente dada por la distancia de los datos con respecto a otros [16], o su producto escalar definido para $\mathbb{R}^p$, algunos ejemplos de las medidas más comunes se describen en la siguiente tabla:

Tabla 1. Ejemplos de distancias y productos escalares comúnmente usados. n denota la dimensión de la data, α_i es el vector de coeficientes y Σ es la matriz de covarianza del data set [16]

Nombre	Distancia	Producto Escalar
Euclidiana	$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$	$\langle x,y\rangle = \sum_{i=1}^{n} x_i\, y_i = x^T y$
Euclidiana ponderada	$d(x,y) = \sqrt{\sum_{i=1}^{n} \alpha_i\,(x_i - y_i)^2}$	$\langle x,y\rangle = \sum_{i=1}^{n} \alpha_i\, x_i y_i$
Mahalanobis	$d(x,y) = \sqrt{(x-y)^t \Sigma^{-1}(x \cdot y)}$	$\langle x,y\rangle = x^T \Sigma^{-1} y$
Minkowski	$d_m = \left(\sum_{i=1}^{n} \|x_i - y_i\|^m\right)^{\frac{1}{m}}$	

Donde:

$x, y \in R^p$

p es la dimensión del data set

$\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_p)$ es un vector de coeficientes positivos

Σ es la matriz de covarianza

Un criterio de selección para la medida de distancia es la distribución de los atributos de los datos, conociendo x_i como un atributo del punto $x = (x_1, x_2, \ldots, x_n)$ se puede especificar una medida de distancia tal que se alcance un criterio de derivabilidad, robustez, u otro criterio definido para la obtención del prototipo.

Las medidas de distancia pueden llegar a presentar valores extremadamente grandes o pequeños, por lo que es necesario normalizar estas medidas de tal forma que se presenten valores relativos a cada categoría en el intervalo $[0.0,1.0]$ y así definir una disimilitud. Dada una distancia d entre los datos x y y, un método de normalización simple de la distancia d consiste en el uso de una transformación lineal dada por:

$$\eta(d) = \frac{d - d_m}{d_M - d_m} \qquad (2.24)$$

Donde:

d_m es la distancia mínima para los datos $Y = y_1, y_2, \ldots, y_n \subseteq R^n$

d_M es la distancia máxima para los datos $Y = y_1, y_2, \ldots, y_n \subseteq R^n$

Esta transformación cumple con el objeto de normalización; para $d = d_m$ tenemos $\eta(d) = 0$ y para $d = d_M$ tenemos $\eta(d) = 1$ siendo $d \in [d_m, d_M]$ donde estos valores de d_m y d_M pueden corresponder a excepciones o datos aberrantes; en estos casos se puede emplear otra normalización definiendo un parámetro $z = (m, M)$, tal que:

$$\eta(d, z) = \min\left(\max\left(\frac{d - m}{M - m}, 0\right), 1\right) \qquad (2.25)$$

El parámetro z se podría identificar como un valor de umbral de tolerancia en el contexto de disimilitud, es decir que para valores $d \leq m$ son considerados totalmente similares, y para valores $d \geq M$ se consideran totalmente disímiles.

2.2.2. Medidas de similitud

Para medir disimilitud podrían haber dos enfoques; el producto escalar para funciones de kernel particulares; o, medidas derivadas de disimilitud a través de funciones decrecientes [20].

2.2.3.2. Producto escalar

El producto escalar definido para funciones de kernel fue introducido por Vapnik, este producto escalar está asociado

implícitamente a transformaciones no lineales de la data, por ejemplo, funciones de kernel polinomial, esto significa que se puede involucrar correlaciones no lineales entre los datos sin aumentar la dimensionalidad del data set y así no incrementar el costo computacional ya que se usa para el cálculo la representación inicial de dicho data set; una función de kernel polinomial puede definirse como:

$$\rho_p(x, y) = (\langle x, y \rangle + l)^\gamma \qquad (2.26)$$

Donde:

$x, y \in \mathbb{R}^p$

p es la dimensión del data set

$\langle x, y \rangle$ es el producto escalar definido en $\mathbb{R}^p$

$l \leq 0$ es un parámetro de compensación

γ es el orden del polinomio

2.2.3.3. Funciones decrecientes de disimilitud

Las funciones decrecientes pueden ser usadas para transformar medidas de disimilitud a medidas de semejanza, ya que las

medidas de semejanza se derivan del complemento a 1 de las medidas de disimilitud [19], tal que:

$$\rho = 1 - \eta_z(d) \qquad (2.27)$$

Donde η_z es la función de normalización para la distancia d. El parámetro z puede diferir para la medida de semejanza $\rho = 1 - \eta_{z_1}(d)$ y la medida de disimilitud $\delta = \eta_{z_2}(d)$ siendo $z_1 \neq z_2$, experimentado diferentes efectos de umbral.

En la siguiente tabla se pueden apreciar las posibles funciones decrecientes para definir semejanza mediante medidas de disimilitud.

Tabla 2. Funciones decrecientes que definen medidas de semejanza [19]

Laplace	$f_l = \frac{1}{a + \left(\frac{d(x,y)}{\sigma}\right)^{\gamma}}$
Gaussiana Generalizada	$f = exp\left(-\left(\frac{d(x,y)}{\sigma}\right)^{\gamma}\right)$
Función Sigmoide o Función Fermi-Dirac	$f_{FD} = \frac{F\left(d(x,y)\right) - F(2\sigma)}{F(0) - F(2\sigma)}; F(z) = \frac{1}{a + exp\left(\frac{z-\sigma}{\gamma}\right)}$

2.2.3. Grados de tipicidad y prototipado difuso

2.2.4.2. Grados de tipicidad

Los grados de tipicidad son medidas que expresan que tan representativo o típico es un elemento en un conjunto; basado en que los miembros de una misma categoría no son necesariamente equivalentes entre sí, ya que algunos miembros son más representativos que otros [18], se dice que un dato es considerado típico si es semejante a los otros miembros de su misma categoría y distinto o disímil de los miembros de otras categorías. Existe un método propuesto por Rifqi [21] para implementar el principio de tipicidad tal que; para cada miembro se calcula su semejanza interna a otros miembros que pertenecen a la misma categoría, y se calcula su disimilitud externa a los puntos pertenecientes a otras categorías; así el grado de tipicidad es la agregación de estas 2 medidas, definida como:

$$R_r(x_i) = \frac{1}{|C_r|} \sum_{y \in C_r} \rho\,(x_i, y) \tag{2.28}$$

$$D_r(x_i) = \frac{1}{|X/C_r|} \sum_{z \notin C_r} \delta\,(x_i, z) \tag{2.29}$$

$$t_{ir} = \begin{cases} \Phi\big(R_r(x_i), D_r(x_i)\big), x_i \in C_r \\ 0, x_i \notin C_r \end{cases} \tag{2.30}$$

Donde:

$X = \{x_1, x_2, \ldots, x_n\}$ es el data set

C_r para $r = 1,2, \ldots, c$ las categorías

ρ la medida de semejanza

δ la medida de disimilitud

$R_r(x_i)$ la semejanza interna en la categoría r para el punto x_i

$D_r(x_i)$ la disimilaridad externa en la categoría r para el punto x_i

Φ es el operador de agregación, tal como una media ponderada o una suma simétrica por ejemplo

t_{ir} el grado de tipicidad en la categoría r para el punto x_i

2.2.4.3. Prototipos difusos

El prototipo es definido como el dato más típico, expresa los puntos más similares en una categoría y también sus características discriminantes con otras categorías [21] [22]; dicho prototipo está dado por la agregación, como una media ponderada, sobre todos los datos de una categoría usando los grados de tipicidad como pesos en la ponderación, formalmente un prototipo se define como:

$$w_r = \frac{\sum_i t_{ir}\, x_i}{\sum_i t_{ir}} \tag{2.31}$$

Donde:

$x_i \in X = \{x_1, x_2, \ldots, x_n\}$

w_r es el prototipo de la categoría r y $r = 1,2,\ldots, c$

t_{ir} el grado de tipicidad en la categoría r para el punto x_i

De forma general se puede definir al prototipo w_r como la agregación de los datos $x_i \in X$ cuyo grado de tipicidad los hace datos representativos [23], como lo siguiente:

$$w_r = \psi(\{x_i / t_{ir} > \tau\}) \tag{2.32}$$

Donde:

ψ es el operador de agregación

τ es el umbral de tipicidad

2.2.4. Interpretación de medias ponderadas

2.3.1.2. Media ponderada

Es el método de obtención de prototipos tradicional, ya que los pesos permiten distinguir unos puntos de otros en la misma categoría aunque generalmente son obtenidos mediante un proceso de medias ponderadas usando como pesos los grados de tipicidad. Para el caso particular de la media aritmética, se considera que es una media ponderada donde todos los peso son iguales; en el caso de la media aritmética se podría definir los prototipos como [16]:

$$w_r = \frac{1}{|C_r|}\sum_{x_i \in C_r} x_i = \min_{z}\left\{\frac{1}{|C_r|}\sum_{x_i \in C_r} |x_i - z|^2\right\} \qquad (2.33)$$

Se debe notar que la media aritmética minimiza una disimilitud externa definida como el punto que maximiza la semejanza interna [16]. Para este caso los datos representativos no se podrían distinguir, puesto que los valores de tipicidad no difieren.

2.3.1.3. Valor más típico

El valor más típico es una medida que puede ser interpretada como la semejanza interna $R_r(x_i)$, reemplazando la similitud promedio al miembro dado de la categoría por la similitud a un

promedio de la categoría [16], depende de la ocurrencia de distancia al centroide estimado, y se define como el punto fijo en la ecuación:

$$f(s) = \frac{\sum_{i=1}^{n} x_i \, f_i(|x_i - s|) m_i^{\lambda}}{\sum_{i=1}^{n} f_i \, (|x_i - s|) m_i^{\lambda}} \qquad (2.34)$$

Donde:

s es el centroide estimado

$x_i \in X = \{x_1, x_2, \ldots, x_n\}$

m_i es el número de ocurrencias del valor x_i

λ es un parámetro definido por el usuario

f_i es una función decreciente que puede ser definida como se muestra en 2.2.2.2.

La definición de Valor más Típico se puede tomar como un caso de media ponderada que incluye una noción de semejanza interna dependiente de la frecuencia de los datos, que se puede expresar para el valor de tipicidad siendo:

$$t_{ir} = f_i(|x_i - s|) m_i^{\lambda} \text{ si } x_i \in C_r \wedge 0 \text{ si } x_i \notin C_r \qquad (2.35)$$

Y el que el término $f_i(|x_i - s|)$ es interpretado como la semejanza interna.

2.3.1.4. Fuzzy C-Means

Fuzzy c-means, como se vio en el punto 2.1.4, es un algoritmo de clustering, el cual puede ser visto como un proceso de medias ponderada de forma iterativa [2], estas medidas ponderadas están dadas en la matriz de membresía $U_r = [u_{ir}]$ donde cada coeficiente u_{ir} se define como [14]:

$$u_{ir} = \left(\sum_{s=1}^{c} \left(\frac{d_{ir}}{d_{is}} \right)^{2/(m-1)} \right)^{-1} ; 1 \leq r \leq c; 1 \leq i \leq n \qquad (2.36)$$

Esta ecuación puede ser llevada a la forma de una transformación, asumiendo como medidas de distancia una norma Euclidiana [16]:

$$u_{ir} = \frac{1}{\sum_{s=1}^{c} \left(\frac{\|x_i - w_r\|}{\|x_i - w_s\|} \right)^{\frac{2}{m-1}}} = \frac{1}{1 + \|x_i - w_r\|^{\gamma} \sum_{s \neq r} \frac{1}{\|x_i - w_s\|^{\gamma}}} \qquad (2.37)$$

Para $\gamma = 2/(m-1)$ y $w = w_1, w_2, \ldots, w_c$ los centroides estimados en las respectivas categorías r.

También, se debe notar que en este proceso los pesos u_{ir} no son igual 0, ya que se mantiene el contexto de que un punto pertenece a más de una categoría a la vez; para el caso en que dos puntos

compartan el mismo valor de membresía supone una dificultad para interpretar estos valores como grados de tipicidad, esto puede indicar que los coeficientes de la matriz de membresía no son semánticamente equivalentes a grados tipicidad ni tampoco a medidas de semejanza, ya que en estas medidas se espera que tenga un comportamiento decreciente con respecto a la distancia a los centroides estimados, pero estos valores son descritos como grados de membresía de un punto a una categoría, esto es el grado de pertenencia hacia una categoría representada como un conjunto difuso [16]. La ecuación que define u_{ir} expresada en términos de γ implica que los coeficientes obedecen a una transformación de Laplace con un $\gamma = 2/(m-1)$, dándole la interpretación de semejanza interna; aunque existe una diferencia con relación a una transformación de Laplace, ya que σ no está definido como una constante y varía con respecto a la distancia de una categoría a cada punto, tal como:

$$\sigma = \frac{1}{\sum_{s \neq r} \frac{1}{\|x_i - w_s\|^{\gamma}}} \qquad (2.38)$$

Donde:

$\|x_i - w_s\|$ es la distancia del punto x_i al centroide estimado de los otros grupos difiriendo con la definición de una transformación de

Laplace como una medida de similitud, a pesar de esto se sugiere que esta transformación no modifica la semántica de la obtención de un prototipo y más bien se podría considerar como una optimización [24].

2.3.1.5. Possibilistic C-Means

Possibilistic C-Means, como se vio en el punto 2.1.5, es una variación del algoritmo de FCM, pero a diferencia de FCM los pesos de membresía dependen del parámetro η_r [15], este parámetro también describe el diámetro de la hiper-esfera del cluster, al igual que FCM se puede considerar como una transformada de Laplace con un $\gamma = 2/(m-1)$ y un σ definido localmente, pero este al contrario de FCM no varía con respecto a la distancia relativa de cada punto a cada centroide estimado ya que $\sigma = \eta_r$; el valor de tipicidad está definido como:

$$t_{ir} = \frac{1}{1 + \left(\frac{\|x_i - w_r\|}{\eta_r}\right)^{2/(m-1)}} \qquad (2.39)$$

Se debe notar que el valor t_{ir} toma valores muy pequeños para los casos en que $\|x_i - w_r\| \ll \eta_r$, e incluso pueden llegar a ser igual a 0.

2.3.1.6. Possibilistic Fuzzy C-Means

Possibilistic Fuzzy C-Means es una combinación de FCM y PCM; en esta variación del algoritmo de clustering difuso el valor de tipicidad se da por la sumatoria de las razones entre la distancia de un punto al centroide y los demás puntos del cluster al centroide; existen dos funciones de distribución de valores de membresía, una de las cuales es similar a la definición de los valores de membresía de FCM, la otra se define de la siguiente forma [25]:

$$t_{ir} = \frac{1}{\sum_{j=1}^{n} \left(\frac{|x_i - w_r|}{|x_j - w_r|} \right)^{2/(m-1)}} \qquad (2.40)$$

Esta definición difiere con FCM en la interpretación del contexto de prototipado difuso para el parámetro $\sigma = \left(\sum_{j=1}^{n} \frac{1}{|x_j - w_r|^{\gamma}} \right)^{-1}$, pues se denota la dependencia de la distancia hacia los otros puntos más no de los centroides estimados en las otras categorías siendo $\gamma = 2/(m-1)$.

2.3. Medidas de validación en algoritmos de clusterizado difuso

Las medidas de validación de clustering permiten determinar el número óptimo de clusters c en un data set, ya que en la mayoría de los casos los algoritmos de clustering requieren como

parámetro este valor; una vez obtenidas las particiones en un proceso de clustering se pueden usar medidas para validar el resultado y así develar, de forma aproximado, la estructura del data set [53]; estas medidas de validación surgen como una necesidad para particiones en conjuntos multidimensionales, pues en estos casos la verificación por visualización de los clusters se vuelve complicada.

En este punto se introducirán las medidas de validación en clustering difuso, las cuales pueden ser usadas para validar un proceso de clustering en los algoritmos revisados en el punto 2.1. Estas medidas de validación se dividen en dos tipo: medidas de validación que dependen de valores de membresía, y medidas de validación que dependen de valores de membresía y el data set; también se revisarán otros enfoques que involucran la validación de clustering difuso.

2.3.1. Medidas de validación que dependen de medidas de membresía

Las medidas de validación que dependen de valores de membresía son aquellas en las que la matriz de partición U es usada para verificar el proceso de clustering, a continuación se

detallan las medidas de validación de clustering difuso que se encuentran en esta categoría.

2.3.1.1. Coeficiente de partición de Bezdek

El coeficiente de partición PC fue introducido por Bezdeck, esta medida se basa en la minimización de las intersecciones entre clusters difusos en la matriz de membresía U [26], este índice de validación se define como:

$$V_{PC} = \frac{1}{n}\sum_{r=1}^{c}\sum_{i=1}^{n}(u_{ir})^2 \qquad (2.41)$$

E indica el promedio relativo entre los valores de membresía, el objeto de este índice es encontrar $\max\{V_{PC}\}; 2 \leq c \leq n-1$ donde c es el número de clusters y n el número de datos en el data set, además se debe considerar que $\frac{1}{c} \leq V_{PC} \leq 1$. Existe una limitación dada para este índice por su tendencia de monotonicidad.

2.3.1.2. Entropía de partición de Bezdek

La entropía de partición PE introducida por Bezdek [26] es una medida escalar definida como una agregación sobre la matriz de membresía dada U tal que:

$$V_{PE} = \frac{-1}{n}\sum_{r=1}^{c}\sum_{r=1}^{n} u_{ir}\log_a(u_{ir}) \tag{2.42}$$

Donde a es la base del logaritmo, el objetivo del índice es encontrar $\min\{V_{PE}\}; 2 \leq c \leq n-1$ dado un número de clusters c y n el número de datos en el data set, considerando $0 \leq V_{PE} \leq \log_a c$. Una limitación conocida para este índice de validación es la monotonicidad que presenta los valores de V_{PE}.

2.3.1.3. Exponente de proporción de Windham

El exponente de proporción de Windham es una medida que utiliza un número máximo de entradas n de una columna en la matriz de partición U, enfocándose en un número de n columnas es posible encontrar grandes valores de maximización local que permitan distinguir subestructuras en el data set [27]. El índice de validación del exponente de proporción se define como:

$$V_{WPE} = \ln\left(\prod_{k=1}^{n}\left(\sum_{j=1}^{u_k^{-1}}(-1)^{j+1}\binom{c}{j}(1-ju_k)^{c-1}\right)\right) \tag{2.43}$$

Donde $u_k = \max_{1\leq i\leq n}\{u_{ik}\}$. Se debe notar que $2 \leq V_{WPE} \leq n-1$.

2.3.1.4. Coeficiente de partición de Dave

El coeficiente de partición de Dave es el Resultado de una modificación de PC [28] para manejar la monotonicidad, esta medida es definida como:

$$V_{MPC} = 1 - \frac{c}{c-1}(1 - V_{PC}) \qquad (2.44)$$

Esta modificación hace que el índice V_{PC} adquiera un comportamiento similar a la medida de descomposición de conjuntos difusos de Backer. Se debe notar que $0 \leq V_{MPC} \leq 1$, y modificando el objetivo para la optimización de c con $\min\{V_{MPC},\}$ siendo $2 \leq c \leq n-1$.

2.3.1.5. Índice de validación de Kim

Este índice de validación está definido por el promedio del grado relativo de intersección entre los $\frac{c}{2}(c-1)$ pares de clusters [29], donde el grado relativo de intersección entre clusters está definido como la suma ponderada de los valores de membresía de x_j hacia cada clúster, El objeto del índice es $\min\{V_{KYI}\}$ tal que $2 \leq c \leq n-1$, este índice se define como:

$$V_{KYI}(U, V: X) = \frac{2}{c(c-1)} \sum_{p \neq q}^{c} \sum_{j=1}^{n} \left(c \cdot \left(u_{Fp}(x_j) \wedge u_{Fq}(x_j) \right) \cdot h(x_j) \right) \qquad (2.45)$$

2.3.1.6. Índice de validación de Chen-Linkens

Este índice de validación se encuentra compuesto de dos términos. El primero refleja la compacidad entre los clusters, para lo cual se usa $\max_{\mathrm{i}}(u_{ik})$ como la medida de distancia relativa de un punto x_k al centroide estimado de un clúster; esto indica que mientras mayor sea este término los datos están mejor clasificados. El segundo término indica la separación entre los clusters, para lo cual la intersección difusa entre los clusters es usada para determinar la separación de estos; mientras más cercano a cera sea la intersección difusa existe una separación más clara de los clusters; en cambio si el valor del primer término es más cercano a $\frac{1}{c}$ entonces x_k tiene valores de membresía más parecidos entre los clusters. El índice de validación combina los conceptos de compacidad y separación entre clusters [20]. El objeto de este índice es el número de clusters c que maximicen V_P, y este se define como:

$$V_P = \frac{1}{n}\sum_{i=1}^{n}\max_{r}\{u_{ri}\} - \frac{1}{k}\sum_{p=1}^{c-1}\sum_{q=p+1}^{c}\left(\frac{1}{n}\sum_{i=1}^{n}\min\left(u_{ip}, u_{jq}\right)\right) \qquad (2.46)$$

Existen varias observaciones sobre este índice:

- Su monotonicidad depende del número de clusters.

- Presenta sensibilidad ante el exponente de ponderación m.
- No usa los datos de forma directa por lo que no relaciona la geometría de la data.

2.3.2. Medidas de validación que dependen de medidas de membresía y el data set

Las medidas de validación que dependen de valores de membresía y el data set son aquellas en las siempre se considera y el data set mismo para verificar el proceso de clustering difuso, a continuación se detallan las medidas de validación que se encuentran dentro de esta categoría.

2.3.3.2. Función de validación de Fukuyama-Sugeno

La función de validación de Fukuyama-Sugeno es una medida que representa qué tan difusos son los valores de membresía en U y la compacidad de la representación geométrica del data set mediante los prototipos; además usa una medida que representa que tan difusa es cada fila en la matriz U y la distancia del prototipo w_r con respecto a la gran media de la data. [31] Este índice se define como:

$$V_{FS} = J_m(u, w) - K_m(u, w) \tag{2.47}$$

$$= \sum_{r=1}^{c}\sum_{i=1}^{n} u_{ir}^{m}\|w_i - w_r\|^2 - \sum_{r=1}^{c}\sum_{i=1}^{n} u_{ir}^{m}\|w_r - \bar{w}\|^2$$

Donde:

$$w = [w_r]; r = 1,2,\ldots,c \tag{2.48}$$

$$\bar{w} = \sum_{r=1}^{c} w_r/c \tag{2.49}$$

El objetivo de la función de validación es $\min\{V_{FS}\}; 2 \leq c \leq n-1$.

2.3.3.3. Función de validación de Xie-Beni

Esta función de validación se enfoca en la separación y compacidad, está dado por la razón de estas dos medidas, siendo el numerador la representación de la compacidad de los valores de membresía, y el denominador representa la intensidad de separación entre los clusters. Esta función establece que para valores óptimos de c se producen valores de compacidad bajos y un alto nivel de separación entre los prototipos w_r [32]. La función se define como:

$$V_{XB} = \frac{J_m\,(u,w)/n}{Sep(w)} = \frac{\sum_{r=1}^{c}\sum_{i=1}^{n} u_{ir}^{m}\|x_i - w_r\|^2}{n \cdot \min\limits_{p,q}\|w_p - w_q\|^2} \tag{2.50}$$

El objetivo de la función de validación es $\min\{V_{XB}\}; 2 \leq c \leq n-1$.

2.3.3.4. Función de validación de Xie-Beni-Kwon

Esta función es una extensión del índice de validación de Xie-Beni para manejar la tendencia de monotonicidad decreciente en casos en que el número de clusters se acerca al número de datos n, para lo cual se agrega un término en el numerador [33], dicha función es definida como:

$$V_k = \frac{\sum_{r=1}^{c}\sum_{i=1}^{n} u_{ir}^{m}\|x_i - w_r\|^2 + \frac{1}{c}\sum_{r=1}^{c}\|w_r - \bar{x}\|^2}{n \cdot \min\limits_{p\neq q}\|w_p - w_q\|^2} \qquad (2.51)$$

Donde:

$$\bar{x} = \sum_{i=1}^{n} x_i/n \qquad (2.52)$$

En el numerador el primer término maneja la similitud entre categorías definido por su compacidad, mientras más símiles son las categorías más pequeño es este término; el segundo término es añadido para eliminar la tendencia decreciente cuando c se acerca a n. El denominador mide la diferencia entre categorías mediante la distancia entre centroides estimados para cada clúster.

El objetivo de esta función de validación es la optimización de c tal que $\min\{V_k\}; 2 \leq c \leq n-1$.

2.3.3.5. Función de validación de Xie-Beni extendida

La función de validación de Xie-Beni extendida es resultado de una modificación sobre la extensión hecha por Kwon a la función de validación de Xie-Beni, esta función es propuesta por Tang [34], en esta el segundo término en el numerador además de manejar la tendencia decreciente del índice dado por la función de validación de Xie-Beni-Kwon también maneja la estabilidad numérica cuando $m \to \infty$, esta función se define como:

$$V_T(U,V;X) = \frac{\sum_{r=1}^{c}\sum_{i=1}^{n} u_{ir}^2 \|x_i - w_r\|^2 + \frac{1}{c(c-1)}\sum_{r=1}^{c}\sum_{p\neq r}^{c}\|w_r - w_p\|^2}{n \cdot \min\limits_{p\neq q}\{\|w_p - w_q\|^2\} + 1/c} \quad (2.53)$$

El objetivo de esta función de validación es la optimización de c tal que $\min\{V_T\}; 2 \leq c \leq n-1$.

2.3.3.6. Función de validación de Zahid

Función propuesta por Zahid quién también propone los conceptos de compacidad difusa y separación difusa [35], los cuales, a diferencia de las medidas de compacidad y separación tradicionales que se enfocan en las propiedades geométricas de los datos; usan la unión difusa y la intersección difusa para el

cálculo de la compacidad y separación difusa respectivamente. Esta función de validación se define por la diferencia entre la compacidad difusa y el grado de separación difusa:

$$V_{SC} = SC_1(c) - SC_2(c) \tag{2.54}$$

Donde:

$$SC_1(c) = \frac{\sum_{r=1}^{c} \|w_r - \overline{w}\|^2 / c}{\sum_{r=1}^{c} (\sum_{i=1}^{n} u_{ir}^m \|x_i - w_r\|^2 / \sum_{i=1}^{n} u_{ir})} \tag{2.55}$$

$$SC_2(c) = \frac{\sum_{r=1}^{c} \sum_{j=r+1}^{n} \left(\sum_{i=1}^{n} \left(\min(u_{ir}, u_{jr}) \right)^2 \Big/ \sum_{i=1}^{n} \min_{1 \le r \le c} (u_{ir}, u_{jr}) \right)}{\sum_{i=1}^{n} \left(\max_{1 \le r \le c} \{u_{ir}\} \right)^2 \Big/ \sum_{i=1}^{n} \max_{1 \le r \le c} \{u_{ir}\}} \tag{2.56}$$

$$\overline{w} = \sum_{r=1}^{c} w_r / c \tag{2.57}$$

El objetivo de esta función de validación es la optimización de c tal que $\max\{V_{SC}\}; 2 \le c \le n - 1$.

2.3.3.7. Coeficiente de validación de hipervolumen de Gath-Geva

Se basa en dos conceptos, hiper-volumen difuso y densidad, Gath y Geva [36] han definiendo el hiper-volumen difuso como:

$$V_{FHV} = \sum_{r=1}^{c} [\det(F_r)]^{1/2} \tag{2.58}$$

Donde la densidad se define como:

$$F_r = \frac{\sum_{i=1}^{n} (u_{ir})^m (x_i - w_r)(x_i - w_r)^T}{\sum_{i=1}^{n} (u_{ir})^m} \tag{2.59}$$

Asumiendo que para valores pequeños de V_{FHV} la partición es considerada estrecha, un óptimo valor de c es dada por $\min\{V_{FHV}\}; 2 \leq c \leq n-1$.

2.3.3.8. Promedio de densidad de Gath-Geva

De la misma forma Gath y Geva definieron la densidad promedio de partición [36] como:

$$V_{APD} = \frac{1}{c} \sum_{r=1}^{c} \frac{S_r}{[det(F_r)]^{1/2}} \tag{2.60}$$

Donde:

c es el número de clusters

$$F_r = \frac{\sum_{i=1}^{n} (u_{ir})^m (x_i - w_r)(x_i - w_r)^T}{\sum_{i=1}^{n} (u_{ir})^m} \tag{2.61}$$

es la densidad, y

$$S_r = \sum_{i=1}^{n} u_{ir}\,; \forall x_i \in \{x_i : (x_i - w_r)\left(F_r^{-1}(x_i - w_r)\right) < 1\} \qquad (2.62)$$

S_r describe la suma de los valores de membresía perteneciente a los datos centrales del clúster, pues estos datos tienen la característica de pertenecer a una región del clúster cercana al centroide estimado w_r. El objetivo del coeficiente es la optimización de c tal que $\max\{V_{DPA}\}; 2 \leq c \leq n-1$.

2.3.3.9. Índice de validación de densidad de Gath-Geva

Gath y Geva propusieron también el índice de densidad de partición [36], el cual depende del valor obtenido por el coeficiente de hiper-volumen difuso, y se define de la siguiente manera:

$$V_{PD} = \frac{\sum_{r=1}^{c} S_r}{V_{FHV}} \qquad (2.63)$$

Donde:

$$S_r = \sum_{i=1}^{n} u_{ir}\,; \forall x_i \in \{x_i : (x_i - w_r)\left(F_r^{-1}(x_i - w_r)\right) < 1\} \qquad (2.64)$$

El objetivo del índice de validación es la optimización de c tal que $\max\{V_{PD}\}; 2 \leq c \leq n-1$.

2.3.3.10. Función de validación de Wu-Yang

Es una función de validación que también permite definir qué tan compacto son los clusters y a su vez qué tan separados se encuentran unos de otros. Dada esta función introducida por Wu y Yang proponen el Coeficiente de Partición y Exponente de Separación [37] y se define como:

$$V_{PCAES} = \sum_{r=1}^{c} PCAES_r = \sum_{r=1}^{c}\sum_{i=1}^{n} u_{ir}^2/\mu_M - \sum_{r=1}^{c} \exp\left(-\min_{p\neq r}\{\|w_r - w_p\|^2/\beta_T\}\right) \quad (2.65)$$

Donde:

$$\mu_M = \min_{1\leq r\leq c}\left\{\sum_{i=1}^{n} u_{ir}^2\right\} \quad (2.66)$$

$$\beta_T = \sum_{r=1}^{c} \|w_r - \bar{x}\|^2/c \quad (2.67)$$

$$\bar{x} = \sum_{i=1}^{n} \frac{x_i}{n} \quad (2.68)$$

El objetivo de la función de validación es la optimización de c tal que $\max\{V_{PCAES}\}; 2 \leq c \leq n-1$.

2.3.3.11. Índice de validación de Tsekouras-Sarimveis

Es una relación entre la compacidad global y la separación difusa; particularmente este índice de validación está definido para cualquier norma-A a diferencia de los índices, coeficientes y funciones de validación vistos hasta ahora; los cuales hacen uso en su mayoría de la norma Euclidiana. Se define la compacidad global como:

$$\pi_r = \frac{\sigma_r}{n_r} = \frac{\sum_{i=1}^{n}(u_{ir})^m \|x_i - w_r\|_A^2}{\sum_{i=1}^{n} u_{ir}}; 1 \leq r \leq c \qquad (2.69)$$

También es definida la separación difusa como:

$$S = \sum_{y}\sum_{z \neq y}\big(dev(y,z,\omega)\big)^2; y,z \in \{w_1, w_2, \ldots, w_c, \bar{x}\} \qquad (2.70)$$

Donde:

$\bar{x} = \sum_{i=1}^{n} \frac{x_i}{n}$ es la gran media sobre el data set

$dev(y,z,\omega) = \big(\mu(z,y,\omega)\big)^{(2+\omega)/\omega}\|z-y\|_A$ es la desviación difusa entre y y z

$\mu(z,y,\omega) = \left[\sum_{v \neq z}\left(\frac{\|z-y\|_A}{\|z-v\|_A}\right)^{\omega}\right]^{-1}; v \in \{w_1, w_2, \ldots, w_c, \bar{x}\}$ es la función de membresía del conjunto difuso $\{w_1, w_2, \ldots, w_c, \bar{x}\}$

Siendo $\omega \in (0, \infty)$ un factor de ajuste definido por el usuario. El índice propuesto por Tsekouras y Sarimveis [38] usa una medida de compacidad para describir la variación de los clusters e introducir el concepto de separación difusa que permite identificar aislamiento de clusters, uno de estos elementos usados para la definición del índice es la desviación difusa entre dos centroides estimados calculada a través de una agregación sobre el resto de centroides y la gran media del data set. El índice de validación de Tsekouras-Sarimveis es definido como:

$$V_{S_{VI}} = \frac{\sum_{r=1}^{c} \pi_r}{S} \tag{2.71}$$

El objetivo del índice de validación es la optimización de c tal que $\min\{V_{S_{VI}}\}; 2 \leq c \leq n-1.$

2.3.3.12. Función de validación de Rezaee-Lelieveldt-Reiber

Es una función que usa la variación promedio en la dispersión entre los clusters y la separación entre la distribución de clusters [39]. Estos dos términos son definidos como dispersión promedio y distancia funcional. La dispersión promedio es definida como:

$$Scat(c) = \frac{\frac{1}{c}\sum_{r=1}^{c} |\sigma(w_r)|}{|\sigma(X)|} \tag{2.72}$$

Donde:

$X \subseteq \mathbb{R}^p$ es el data set

$\sigma(X) \in \mathbb{R}^p$ es la varianza del data set X y es definida como:

$$\sigma(X) = \left(\sigma_1(X), \sigma_2(X), \ldots, \sigma_p(X)\right); \sigma_k(X) = \frac{1}{n}\sum_{i=1}^{n}\left(x_{i_k} - \bar{x}_k\right)^2; 1 \leq k \leq p \quad (2.73)$$

$\bar{x} = \sum_{i=1}^{n} x_i/n$ es la gran media sobre el data set

$\sigma(w_r) \in \mathbb{R}^p$ es la variación difusa sobre el clúster r y se define como:

$$\sigma(w_r) = \left(\sigma_1(w_r), \sigma_2(w_r), \ldots, \sigma_p(w_r)\right); \quad (2.74)$$

$$\sigma_k(w_r) = \frac{1}{n}\sum_{i=1}^{n} u_{ir}\left(x_{i_k} - w_{r_k}\right)^2; 1 \leq k \leq p$$

La distancia funcional es definida como:

$$Dis(c) = \frac{D_{\max}}{D_{\min}}\sum_{r=1}^{c}\left(\sum_{q=1}^{c}\|w_r - w_q\|\right)^{-1} \quad (2.75)$$

Donde:

$D_{\max} = \max\{\|w_p - w_q\|\}; p \neq q \wedge w_p, w_q \in w_1, w_2, \ldots, w_c$ es la distancia máxima entre los centroides estimados

$D_{\min} = \min\{\|w_p - w_q\|\}; p \neq q \wedge w_p, w_q \in w_1, w_2, \ldots, w_c$ es la distancia mínima entre los centroides estimados

Dadas estas dos medidas Rezaee, Lelieveldt y Reiber definen el índice de validación como:

$$V_{CWB} = \alpha Scat(c) + Dis(c) \tag{2.76}$$

Donde:

$\alpha = Dis(c_{\max})$ es la distancia funcional sobre el número máximo de clusters $c_{\max}$

En el primer término se asume que a medida que la dispersión entre clusters aumenta estos son menos compactos, dado que este término es un indicador de compacidad. El segundo término describe la separación total de dispersión entre clusters, este aumenta dependiendo del número de clusters c y su comportamiento es determinado por la geometría de los clusters. El peso α es usado en el primer término para ponderar esta medida, dado que los dos términos se manejan en rangos diferentes. El objetivo de la función de validación es la optimización de c tal que $\min\{V_{CWB}\}; 2 \leq c \leq n - 1$.

2.3.3.13. Función de validación de Sun

Es una función de validación introducida por Sun basada en el trabajo de Rezaee, Lelieveldt y Reiber, define una función de validación modificando el segundo término del índice [40], definiendo un término representativo de la separación entre clusters, dicha medida es definida como:

$$Sep(c) = \frac{D_{\max}^2}{D_{\min}^2} \sum_{r=1}^{c} \left(\sum_{q=1}^{c} \|w_r - w_q\|^2 \right)^{-1} \tag{2.77}$$

Y el índice de validación se define como:

$$V_{WSI} = Scat(c) + \frac{Sep(c)}{Sep(c_{\max})} \tag{2.78}$$

El objetivo de la función es la optimización de c tal que $\min\{V_{WSI}\}; 2 \leq c \leq n-1$.

2.3.3.14. Función de validación de Kim

Esta función es definida por la razón entre el grado de traslape y el grado de separación [41]. El grado de traslape es definido de la siguiente forma:

$$Overlap(c, U) = \frac{2}{c(c-1)} \sum_{p=1}^{c-1} \sum_{q=p+1}^{c} \left[\sum_{u} \sum_{i=1}^{n} \delta\left(x_i, u: \tilde{F}_p, \tilde{F}_q\right) \omega(x_i) \right] \tag{2.76}$$

Donde:

$\tilde{F}_p$ y $\tilde{F}_q$ son conjuntos difusos tales que $\tilde{F}_r = \left\{\left(x_r, u_{\tilde{F}_r}(x_r)\right) \vee x_r \in X\right\}$ siendo X el data set

u es el valor de membresía asociado al punto x_i

$$\delta\left(x_i, u: \tilde{F}_p, \tilde{F}_q\right) = f(x) = \begin{cases} \omega(x_i) & \text{,si } \left(u_{\tilde{F}_p}(x_i)\right) \geq u \wedge \left(u_{\tilde{F}_q}(x_i)\right) \geq u \\ 0 & \text{,caso contrario} \end{cases} \qquad (2.79)$$

$\omega(x_i) \in [0.0,1.0]$ es un peso definido para valores de membresía dentro de un rango dado tal que:

$$\omega(x_r) = \begin{cases} \omega_1, u_{\tilde{F}_r}(x_r) \in [t_0, t_1] \\ \omega_2, u_{\tilde{F}_r}(x_r) \in [t_1, t_2] \\ \vdots \\ \omega_m, u_{\tilde{F}_r}(x_r) \in [t_{m-1}, t_m] \end{cases} \qquad (2.80)$$

Este peso es determinado por el grado de traslape del punto x_r entre los clusters a los que pertenece. El grado de separación es definido de la siguiente forma:

$$Sep(c, U) = 1 - \min_{p \neq q}\left\{\max_{x \in X}\left\{\min\left(u_{\tilde{F}_p}(x), u_{\tilde{F}_q}(x)\right)\right\}\right\} \qquad (2.81)$$

Este indica el grado de aislamiento de la distancia entre los clusters difusos. Se define la función de validación propuesta por Kim como:

$$V_{OS} = \frac{Overlap^N(c,U)}{Sep^N(c,U)} \quad (2.82)$$

Donde:

$Overlap^N(c,U) = \frac{Overlap(c,U)}{Overlap_{\max}}$ es el grado de traslape normalizado

$Sep^N(c,U) = \frac{Sep(c,U)}{Sep_{\max}}$ es el grado de separación normalizado

El objetivo de la función de validación es la optimización de c tal que $\min\{V_{OS}\}; 2 \leq c \leq n-1$.

2.3.3.15. Índice de validación de Pakhira

Es un índice desarrollado para clustering difuso, el cual usa tres términos para su definición; el primero define la divisibilidad de los c clusters, el segundo es una medida de compacidad dada por la razón entre la suma ponderada de las distancias desde cada punto en el data set al primer clúster, y la agregación de la distancia ponderada desde los puntos del data set hacia cada clúster; el tercer término mide la separación máxima entre cada clúster [42]. Este índice es definido como:

$$V_{PMBF} = \left(\frac{1}{c} \times \frac{E_1}{J_m} \times D_c\right)^2 \quad (2.83)$$

Donde:

$$E_r = \sum_{i=1}^{n} u_{ir} \|x_i - w_r\| \tag{2.84}$$

$$D_c = \max_{p,q \in [1.0,c]} \|w_p - w_q\| \tag{2.85}$$

$$J_m = \sum_{i=1}^{n} \sum_{r=1}^{c} (u_{ir})^m \|x_i - w_r\| \tag{2.86}$$

Se puede observar que el primer factor tiende en decremento cuando c aumenta; también notamos que en el segundo factor el denominador es decreciente cuando c aumenta mientras que el numerador es un valor fijo, y el factor final depende de la separación de los clusters. Siendo así, el objetivo del índice de validación la optimización de c tal que $\max\{V_{PBMF}\}; 2 \leq c \leq n-1$.

2.3.3.16. Función de validación de Bouguessa-Wang

Esta función de validación usa una definición de compacidad global dependiente de una covarianza difusa; también se define una función de separación difusa entre clusters [43]. Así se define la compacidad global como:

$$GComp(c) = \sum_{r=1}^{c} trace(\Sigma_r) \tag{2.87}$$

Donde:

$\Sigma_r = \frac{\sum_{i=1}^{n} u_{ir}^{m}(x_i - w_r)(x_i - w_r)^T}{\sum_{i=1}^{n} u_{ir}^{m}}$ es la matriz de covarianza difusa

La separación difusa es definida como:

$$S(c) = trace(B) \tag{2.88}$$

Donde:

$S_B = \sum_{r=1}^{c} \sum_{i=1}^{n} u_{ir}^{m}(w_r - \overline{w})(w_r - \overline{w})^T$ es la matriz de separación entre clusters

$\overline{w} = \sum_{r=1}^{c} w_r / c$ es la gran media sobre los centroides estimados

Finalmente se define la función de validación como la razón de estas dos medidas como:

$$V_{SCG} = \frac{S(c)}{GComp(c)} \tag{2.89}$$

Un valor grande para el numerador indica que las particiones difusas están bien separadas y para valores pequeños del denominador se indican particiones compactas. Particiones difusas compactas y separadas de forma equilibrada pertenecen a valores grandes de V_{SCG}. El objetivo de la función de validación es la optimización de c tal que $\max\{V_{SCG}\}; 2 \leq c \leq n - 1$.

2.3.3.17. Índice Dunn-Hassar-Hensaid

Es un índice generalizado derivado del índice propuesto por Dunn para conjuntos no difusos [44] definido como:

$$V_D = \min_{1\le i\le k}\left\{\min_{1\le j\le K, j\neq i}\left\{\frac{\delta(C_i, C_j)}{\max\limits_{1\le k\le K}\Delta(C_k)}\right\}\right\} \tag{2.90}$$

Para este índice se definen las medidas de distancia entre clusters difusos, la cual es usada en el numerador, y diámetro de clúster difuso derivadas de las definiciones para datos no difusos. Hassar y Hensaid proponen varias opciones para la medida de distancias entre clusters difusos:

$$\delta^{fuzzy}(S,T) = \delta^{fuzzy}_{average}(S,T) = \frac{1}{n_S n_T}\sum_{x,y\in X} u_S(x) * u_T(y) * d(x,y) \tag{2.91}$$

$$\delta^{fuzzy}(S,T) = d\left(w_S^f, w_T^f\right) \tag{2.92}$$

$$\delta^{fuzzy}(S,T) = \delta^{fuzzy}_{average}(S,T) = \frac{1}{n_S n_T}\sum_{x,y\in X} u_S(x) * u_T(y) * d(x,y) \tag{2.93}$$

$$\delta^{fuzzy}(S,T) = \frac{1}{n_S + n_T}\left[\sum_{x\in X} u_S(x) * d\left(x, w_S^f\right) + \sum_{y\in Y} u_T(y) * d\left(y, w_T^f\right)\right] \tag{2.94}$$

$$\delta^{fuzzy}(S,T) = \delta^{fuzzy}_{Hausdorff}(S,T) = max\{\delta^f(S,T), \delta^f(T,S)\} \tag{2.95}$$

Donde:

$$\delta^f(S,T) = \max\left\{u_S(x), \min_{\substack{y\in X \\ x\neq y}}\left\{\frac{d(x,y)}{u_T(y)}\right\}\right\} \quad (2.96)$$

Y

$$\delta^f(T,S) = \max\left\{u_T(x), \min_{\substack{y\in X \\ x\neq y}}\left\{\frac{d(x,y)}{u_S(y)}\right\}\right\} \quad (2.97)$$

La medida de diámetro de clusters difusos es definida como:

$$\Delta^{fuzzy}(S) = 2\cdot\left\{\sum_{x\in X}\frac{u_S(x)\cdot d\left(x, w_S^f\right)}{n_S}\right\} \quad (2.98)$$

Donde:

$n_S = |X_S| = \sum_{k=1}^{n} u_S(x_k)$ es la cardinalidad del clúster S

$w_S^f = \frac{1}{n_S}\sum_{x_k\in X} u_S(x_k)x_k$ es el centroide estimado o el vector de media difusa del clúster S

De forma general el objetivo del índice de validación es la optimización de c tal que $\min\{V_D\}; 2 \leq c \leq n-1$.

2.3.3.18. Índice de granularidad-disimilitud de Xie

Este índice de validación está definido bajo la combinación de las medidas de disimilitud y granularidad para clusters [45]. Dado los

clusters $C = \{C_1, C_2, \dots, C_c\}$ en un data set $X = \{x_1, x_2, \dots, x_n\}$, tal que $C' = \{C_{pi} | (C_p \in C), i = 1,2, \dots, k$ donde $k = |C'|\}$ se define el grado de granularidad GR para los clusters C como:

$$GR = \frac{\sum_{r=1}^{k} \frac{\sum_{i=1}^{n} (u_{ir})^2 d(x_i, w_r)}{\sum_{i=1}^{n} (u_{ir})^2}}{k} \quad (2.99)$$

De la misma forma se define la medida de disimilitud DS como:

$$DS = \left(\frac{\sum_{r=1}^{c} \min_{1 \leqslant q \leqslant c, r \neq q} d(w_r, w_q)}{c} \right)^2 \quad (2.100)$$

Estos dos conceptos son usados para definir el índice de validación de granularidad-disimilitud como:

$$V_{GD} = \frac{k}{c} \times \frac{DS}{GR} \quad (2.101)$$

El objetivo del índice de validación es la optimización de c tal que $\max\{V_{GD}\}; 2 \leqslant c \leqslant n - 1$.

2.3.3.19. Índice de validación difusa de Yu-Li

Yu y Li proponen un índice para medir la estabilidad de un clúster difuso producto de FCM dada por la estabilidad de la matriz Hessiana H_v mediante el número de condición en dicha matriz [46], pues este describe estabilidad mediante la razón de valores

evaluados en la matriz Hessiana, usando los centroides estimados de los clusters producto de FCM. Así se define la matriz Hessiana como:

$$H_v = \frac{\delta^2 F(w)}{\delta w_p \delta w_q} \tag{2.102}$$

$$= \frac{4m}{m-1} \sum_{i=1}^{n} {u_{ip}}^{(m+1)/2} {u_{iq}}^{(m+1)/2} \frac{(x_i - w_p)(x_i - w_q)^T}{\|x_i - w_p\| \|x_i - w_q\|}$$

$$+ 2\delta_{pq} \left[\left(\sum_{j=1}^{n} {u_{jp}}^{m} \right) \times I_{S \times S} \right.$$

$$\left. - \frac{2m}{m-1} \sum_{i=1}^{n} {u_{ip}}^{m} \frac{(x_i - w_p)(x_i - w_p)^T}{\|x_i - w_p\|^2} \right]; p, q \in [1.0, c]$$

Así también se define el número de condición para la matriz Hessiana H_v como:

$$V_{cond(H_v)} = \frac{\lambda_{\min}(H_v)}{\lambda_{\max}(H_v)} \tag{2.103}$$

Donde:

$\lambda_{\min}(H_v)$ es el valor propio mínimo para H_v

$\lambda_{\max}(H_v)$ es el valor propio máximo para H_v

Si $cond(H_v) < 0$ el clustering se presenta inestable, si $cond(H_v) > 1$ la estabilización del clustering se presenta imposible; para $0 <$

$cond(H_v) < 1$ mientras más se acerque a uno el clustering presenta mayor estabilidad. El objetivo del índice de validación $V_{cond(H_v)}$ es la optimización de c tal que $\max\{V_{cond(H_v)}\}; 2 \leq c \leq n - 1$.

2.3.3. Otros enfoques de índices de validación

Las medidas de validación basadas en otros enfoques no usan medidas de compacidad y separación como las medidas de validación que dependen de valores de membresía, y las medidas de validación que dependen de valores de membresía y el data set, pues estos índices hacen uso de las distancias entre los centroides, teoría de Bayes o medidas de fusión para validar el proceso de clustering; a continuación se detallan tres enfoques de índices de validación que no hacen uso de las medidas de compacidad o separación.

2.3.3.1. Distancia entre centroides

Se basa en que los valores de membresía tienden a ser máximos por el incremento en el número de clusters c cuando se acerca al número de datos n [47] [48]. Esta validación está dada por:

$$\lim_{c \to n} \sum_{r=1}^{c} \sum_{i=1}^{n} (u_{ir})^m \|x_i - w_r\|^2 = 0 \tag{2.104}$$

2.3.3.2. Puntuación Bayesiana de Cho-Yoo

Este método es denominado validación Bayesiana difusa [50], es inspirado por el concepto clásico de teoría de probabilidades [49], para una partición difusa con valores de membresía máximos dado para un data set se tiene:

$$\max P(Cluster|Dataset) \tag{2.105}$$

Definiéndose la puntuación Bayesiana como:

$$BS = \frac{\sum_{r=1}^{c} P(c_r/D_r)}{c} = \frac{\sum_{r=1}^{c} \prod_{i}^{N_r} P(c_r)P\,(d_{ir}/c_r)/P\,(d_{ir})}{c} \tag{2.106}$$

Donde:

$$D_r = d_{ir} \vee u_{ir} > \alpha, 1 \leq i \leq n \tag{2.107}$$

$$N_r = n(D_r) \tag{2.108}$$

$$P(c_r) = \frac{\sum_{i=1,u_{ir}>\alpha}^{n} u_{ir}}{\sum_{r=1}^{c}\sum_{i=1}^{n} u_{ir}} \tag{2.109}$$

$$P(d_{ir}) = \sum_{r=1}^{c} P(C_r)P(d_{ir}/C_r) = \sum_{r=1}^{c} P(C_r)u_{ir} \tag{2.110}$$

Dado que el valor de membresía u_{ir} representa la pertenencia del punto x_i a un clúster C_r, u_{ir} puede ser reemplazado por $P(d_{ir}/C_r)$.

2.3.3.3. Optimización de número de clusters de Rhee-Oh

Rhee y Oh proponen una función de validación para la optimización de clustering difuso [51] dada por las medidas de compacidad total y la separación de las particiones difusas; siendo un número n de datos en un data set X, y u_{ir} el valor de membresía del punto x_i perteneciente al clúster C_r se definen:

$$C = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \sum_{r=1}^{c} d(x_i, x_j)^2 \omega \qquad (2.111)$$

Como la distancia total entre clases, dada por la agregación de la distancia entre los puntos pertenecientes a un clúster C_r, donde $\omega = \min\{u_{ir}, u_{jr}\}$. También se define:

$$D = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} d(x_i, x_j)^2 \varepsilon \qquad (2.112)$$

Como la distancia total entre clusters dada por las distancias entre los puntos pertenecientes al clúster C_p y los puntos pertenecientes al clúster C_q, tal que $p \neq q$ y $\varepsilon = \min\left\{\max_p\{u_{ip}\}, \max_{p \neq q}\{u_{iq}\}\right\}$.

La función de validación de compacidad-separación está dada por la razón de las medidas antes mencionadas, definida como:

$$G = \frac{D}{C} \tag{2.113}$$

Luego definimos la medida de disimilitud entre las poblaciones en cada clúster como:

$$P = 1 - \frac{\sigma}{\sigma_{\max}} \tag{2.114}$$

Donde σ es la desviación estándar de $N = \{n_1, n_2, \dots, n_c\}$ como las cardinalidades de los clusters siendo $n_r = |C_r|$, y $\sigma_{\max}$ es la desviación estándar de $\acute{N} = \{n, 0, \dots, 0\}$.

Dada la función de validación de compacidad-separación y la medida de disimilitud entre poblaciones se define el índice de validación dado por la medida de compacidad total y separación como:

$$I_G = \frac{G}{f(P)} \tag{2.115}$$

Donde:

$f(P) = P^z; z = \frac{1}{2}, 1, 2, \dots$ siendo z un parámetro definido por el usuario

Dado esto se define en el contexto de optimización del número de clusters c el grado de validez de clusters CV tal que:

$$CV(c) = \frac{1}{2}[R(c) + A(c)] \qquad (2.116)$$

Donde:

$A(c) = \frac{I_G(c)}{\max\limits_{2 \le c \le n-1}\{I_G(c)\}}$ es el valor normalizado de $I_G(c)$

$R(c) = H\big(\alpha(\pi_1 + \pi_2)\big)$ es la función no lineal H que define la cantidad relativa de cambio en c

Para $\pi_1 = \frac{I_G(r) - I_G(r-1)}{I_G(r)}$ y $\pi_2 = \frac{I_G(r) - I_G(r+1)}{I_G(r)}$, se puede considerar π_1 y π_2 decrecientes con respecto a c siendo $\alpha = \frac{c-1}{c}$. El objetivo del grado de valides de clusters es la optimización de c tal que $\max\{CV(c)\}; 2 \le c \le n-1$.

2.3.3.4. Índice de validación Fukuyama-Sugeno con índice de fusión

Es una técnica híbrida que usa un índice de validación difuso de clusters, propuesto por Fukuyama y Sugeno [31], y el índice de fusión para optimización del número de clusters c en un data set [52]. El índice de fusión es definido como:

$$P(w) = \sum_{j=1}^{n} exp\left(-4\left|\frac{(w - x_j)/(w_i - w_j)}{2}\right|^2\right) \qquad (2.117)$$

Donde:

$\frac{w_i - w_j}{2}$ es el radio de influencia

Basado en el criterio de fusión, para dos centroides estimados w_i y w_j pueden ser fusionados si para el punto medio $w_m = \frac{(w_i + w_j)}{2}$ el vlaor de $P(w_m)$ es mayor que $P(w_i)$ y menor que $P(w_j)$ caso contrario se mantienen en clusters sin fusionarse.

La técnica híbrida consta de dos pasos: en el primero es usando un índice de validación de clusters FS para estimar un primer número óptimo de clusters; luego en el segundo paso este número es refinado para obtener una mayor precisión con respecto al número de clusters con procesos de índice de fusión.

2.4. Estimación de medidas en contexto académico

Las medidas en contexto académica permitirán abstraer desde la historia académica de un estudiante sus características y las características de las materias que cursa, estas características serán usadas en el proceso de clustering, ya que describen la carga académica semestral y el rendimiento de un estudiante.

2.4.1. Medida de dificultad

Esta es una medida introducida por Caulkins que representa, para un determinado curso j, el nivel de dificultad de éste como una

razón entre la sumatoria del Promedio de Calificaciones (Grade Point Average GPA) al cuadrado, para cada estudiante i que haya tomado dicha materia j; y la sumatoria de la multiplicación entre la calificación del estudiante en la materia j y el GPA del mismo estudiante [6]. Formalmente se define como:

$$\alpha_j = \frac{\sum_i GPA_i^2}{\sum_i (r_{ij} \cdot GPA_i)} \tag{2.118}$$

Donde:

GPA_i es el Promedio de Calificaciones del estudiante i

r_{ij} es la calificación del estudiante i en la materia j

Esta medida está dada para una materia, su valor está en función del número de estudiantes que han la han tomado y un vector de características dado por el estudiante $e_{ij} = (GPA_i, r_{ij})$ relacionado a la materia j.

2.4.2. Medida de rigurosidad

De igual manera propuesta por Caulkins, esta medida describe la rigurosidad con que es calificada una materia j; está definida por la sumatoria de la distancia entre el GPA del estudiante i y la

calificación obtenida por este en la materia j [6]. Se define formalmente como:

$$\beta_j = \frac{\sum_i (GPA_i - r_{ij})}{N_S^j} \quad (2.119)$$

Donde:

GPA_i es el Promedio de Calificaciones del estudiante i

r_{ij} es la calificación del estudiante i en la materia j

N_S^j es el número de estudiantes que han tomado la materia j

Es una medida definida para una materia e igualmente que la dificultad depende del número de estudiantes y el vector de características relacionado a cada uno de estos; además esta medida describe el desplazamiento producto de la influencia en el GPA que tiene la calificación obtenida por el estudiante i en la materia j.

2.4.3. Medida de distribución de notas

Es una medida introducida por Méndez [7], la cual describe la distribución de la distancia entre la calificación obtenida por el estudiante i en la materia j y el GPA del estudiante; dado esto para distribuciones con una asimetría positiva se puede expresar que la

calificación del estudiante en la materia j desplaza su GPA de forma negativa, mientras que para distribuciones con asimetría negativa el GPA es menor que la calificación del estudiante en la materia j por lo que su GPA es desplazado de forma positiva, finalmente para distribuciones de tendencia simétrica se podría expresar que la influencia de la calificación en la materia sobre el GPA del estudiante es casi nula. La medida de distribución de notas se define como:

$$Sk_j = skewness_i\left(GPA_i - r_{ij}\right) \tag{2.120}$$

Donde:

GPA_i es el Promedio de Calificaciones del estudiante i

r_{ij} es la calificación del estudiante i en la materia j

2.4.4. Estimación de dependencia entre medidas

En una malla curricular tomada como un conjunto de materias, es posible definir si existen categorías o clusters de estas materias; Méndez propone que en un análisis de relación entre materias de la misma malla curricular, como una simple medida de grado de dependencia, es posible usar un coeficiente de correlación de Pearson sobre las calificaciones de los estudiantes que han

tomado un par de materias para estimar su relación lineal [7], pero concluye que no hay una correlación significativa entre la mayoría de las calificaciones de los estudiantes, dejando abierta la posibilidad de un estudio de relación con factores no explorados en dicho análisis. En este trabajo se establece una estimación de dependencia entre medidas que permita definir una relación para materias de una misma categoría; para lo cual se definirá un vector de características para cada materia en una malla curricular y así poder estimar mediante técnicas de clustering la relación entre las medidas establecidas en dicho vector, así tenemos que para una materia m_j se establece un vector de característica definido como:

$$m_j = (\alpha_j, \beta_j, Sk_j) \tag{2.121}$$

Donde:

α_j es la medida de dificultad para la materia j

β_j es la medida de rigurosidad para la materia j

Sk_j es la medida de distribución de notas

2.4.5. Exploratory Factor Analysis

Bajo la hipótesis de la existencia de estructuras coherentes de agrupación de materias en una malla curricular; Méndez propone el

uso de Exploratory Factor Analysis (EFA) para develar estas estructuras [7], concluye que los factores resultantes describen grupos de materias interrelacionadas llamados factores, este análisis fue realizado para estudiantes graduados entre 2000 y 2012 de Ingeniería en Ciencias Computacionales y sus equivalentes carreras predecesoras, para este trabajo se usarán estos factores para describir medidas en las habilidades adquiridas a través de la carrera por un estudiantes de la carrera de Ingeniería en Ciencias Computacionales, se podría generalizar que los factores aportan al desarrollo de habilidades requeridas a nivel profesional y que esta medida se encuentra en función de las calificaciones obtenidas en las materias que describen un factor. Los factores que fueron definidos son cinco, y se describen de la siguiente forma:

1. *Factor de preparación en Ingeniería Básica*. Las materias pertenecientes a este factor son clasificadas en dos sub-grupos: ciencias básicas, y materias de ciencias computacionales fundamentales. El primer sub-grupo permite el desarrollo de la habilidad para entender conceptos e ideas abstractas, como matemáticas, lógica, fenómenos naturales e interpretación de fenómenos estadísticos. El segundo sub-grupo permite el desarrollo de habilidades relacionadas a la

lógica, matemática y análisis de datos desde un enfoque netamente computacional.

2. *Factor de Interacción con el Cliente*. Relacionado a la habilidad de comunicación con el usuario final y clientes parte del proceso de desarrollo de software.

3. *Factor de Temas Avanzados de Ciencias Computacionales*. Está conformado por materias que describen el desarrollo de la habilidad para comprender e interpretar conceptos relacionados a Ciencias Computacionales a un alto nivel, como por ejemplo Ingeniería de Software, Organización y Arquitectura de Computadores, Interacción Humano-Computadora o Inteligencia Artificial. En general facultan la habilidad para desarrollar, diseñar e identificar los componentes de un sistema y su relación.

4. *Factor de Programación*. En este factor se encuentran agrupadas materias que ayudan al desarrollo de las habilidades de programación, e implementación así como conceptos, estrategias y patrones de diseño usados en el proceso de desarrollo de software.

5. *Factor de Cursos No Estrechamente Relacionados a Ciencias Computacionales*. Se encuentra formado por materias

relacionadas en su mayoría a ingeniería eléctrica y más no a solucionar problemas relacionados a las Ciencias Computacionales.

Dada estos factores, se establece una medida que describa las habilidades adquiridas mediante el desempeño en las materias relacionadas a este factor, esta medida está dada por; el número de materias perteneciente a un factor F_k en las que un estudiante i se ha registrado, el número de materias perteneciente a un factor F_k en las que el estudiante i se ha registrado y las ha aprobado, y el promedio de las calificaciones obtenidas en las materias pertenecientes al factor F_k. Esta medida está definida como:

$$f_{k_i} = \frac{n_{a_k}}{n_{e_k}} \cdot \sum_{j \in F_k} \frac{r_{ij}}{n_{e_k}} \qquad (2.122)$$

Donde:

n_{a_k} es el número de materias aprobadas por el estudiante i que pertenecen al factor F_k

n_{e_k} es el número de materias tomadas por el estudiante i que pertenecen al factor F_k

r_{ij} es la calificación del estudiante i en la materia j perteneciente al factor F_k

Esta medida está definida en términos de eficiencia para aprobar materias y las calificaciones obtenidas por un estudiante dentro de un factor dado; pues aunque la agregación de las calificaciones en el factor describe en cierta medida el rendimiento de un estudiante, este no es proporcional con respecto al número de materias en las que este estudiante tiene éxito, por esta razón el término $\frac{n_{a_k}}{n_{e_k}}$ está presente en la definición de la medida.

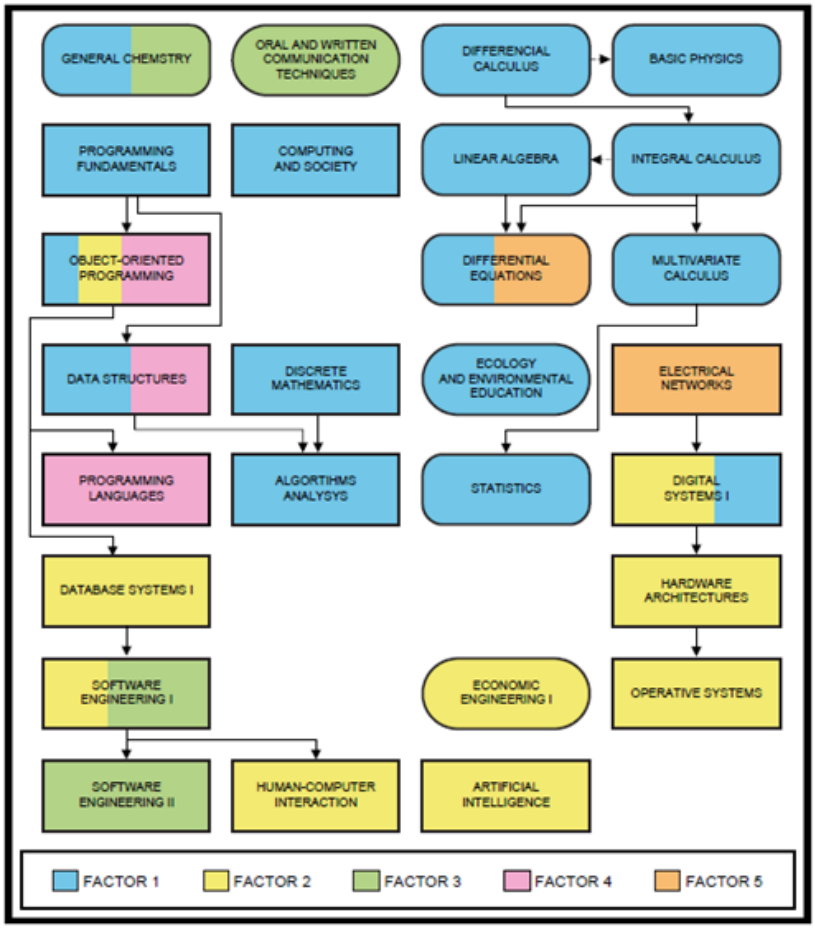


Figura 2.1. Materias categorizadas de acuerdo al factor al que pertenecen [7]

2.4.6. Medida de carga académica

Es una medida relacionada a un semestre en específico y un estudiante en específico. Puede ser descrita como una medida asociada a diferentes variables como el número de materias tomadas por un estudiante dado en un semestre, la sumatoria sobre las medidas de dificultad de las materias tomadas en un semestre, la sumatoria de las medidas de rigurosidad en las materias tomadas en un semestre[7].

n_{H_t} como el número de materias tomadas por un estudiante en un semestre H_t

$\alpha_S = \sum_{j \in H_t} \alpha_j$ como el total de dificultad a la que se afronta un estudiante en el semestre H_t

$\beta_S = \sum_{j \in H_t} \beta_j$ como la rigurosidad total a la que es sometido un estudiante en el semestre H_t

2.4.7. Medida de rendimiento académica

Estas medidas se encuentran relacionadas a un estudiante dado y describen su rendimiento relativo en función de las medidas sobre los cinco factores mencionados en el punto 2.4.5. Cada uno de estos factores describen de forma independiente el desarrollo de

las habilidades requeridas para aprobar las materias interrelacionadas dentro del factor sobre el que son medidas, además dichas medidas no dependen del nivel al que un estudiante pertenezca pero permiten diferenciar a estudiantes según su eficiencia aprobando materias [7], es decir, si tomamos como ejemplo dos estudiantes, el primero solo ha tomado una materia relacionada al Factor 1 pero la ha reprobado con éxito teniendo una calificación de 7.0, mientras que el segundo ha tomado tres materias del Factor 1, debido a que está en un nivel avanzado, aprobándolas con éxito con las calificaciones de 8.0, 6.0 y 7.0 respectivamente, sus habilidades relacionadas al Factor 1 están desarrolladas al mismo nivel sin importar cuantas materias han tomado ya que todas las materias dentro de ese factor requieren la misma habilidad para ser aprobadas; ahora si tomamos otro ejemplo, dos estudiantes han tomado dos materias dentro del mismo factor, el primer estudiante aprobó con éxito las dos materias con 7.0 y 7.0 como calificaciones, mientras que el segundo estudiante aprobó una materia con la calificación de 10.0 mientras la segunda materia la reprobó con 4.0, si solo se considera las calificaciones los dos estudiantes tendría un puntaje de 7.0 en dicho factor, pero es notorio que al reprobar una materia el segundo no está en el mismo nivel de desarrollo de la habilidad

requerida para aprobar las materias dentro del factor, por lo que una calificación más real para describir el rendimiento de un estudiante también debe de encontrarse en función de la eficiencia de un estudiante para aprobar las materias.

2.5. Comparación de variantes de métodos de clusterizado

En esta sección se realizará un análisis comparativo entre las técnicas de clusterizado y clusterizado difuso revisadas anteriormente, para el objeto de comparación se tomarán en cuenta los siguientes criterios: parámetros utilizados, escalabilidad y casos de uso; el último de estos criterios es muy importante en la explicación del diseño de la arquitectura.

K-means es un método cuyos parámetros de entrada son el data set y el número de clusters que se espera como resultado, posee una escalabilidad de n muestras para un número de clusters n; su uso fundamental es amplio puesto que es un algoritmo de categorización de propósito general, pero generalmente es usado conociendo el número de clusters previamente, además que por lo general los clusters resultantes se presentan en geometrías planas puesto que como método de comparación es usada la distancia entre puntos.

El método de ***propagación por afinidad*** tiene como parámetros el data set, el parámetro de afinidad, medida de distancia y la muestra de preferencias, el número de clusters no puede ser escalado ya que es determinado en el proceso de clustering; y generalmente su uso es dado en clasificaciones extensas donde existe la posibilidad de encontrar muchos clusters formados por conjuntos de datos de un tamaño mucho menor a la muestra, aunque estos clusters tienen la característica de no necesariamente presentarse como geometrías planas por la ventaja de poder representar la data como un grafo.

El ***clustering aglomerativo*** tiene como parámetros el número de clusters, la medida de distancia a utilizar, el tipo de vínculo y el data set; su escalabilidad es dada para n muestras a través de c clusters, su uso es por lo general para determinar grupos relacionados mediante niveles jerárquicos, dado que los clusters posiblemente se encuentran relacionados entre ellos, además tienen la característica de poder trabajar con distancias no euclidianas, pues la medida de distancia puede ser cualquier función tal que $f(x, y) = z$ siendo $x, y \in X \subseteq \mathbb{R}^p$ pertenecientes al data set.

El algoritmo de ***FCM*** tiene como parámetros además del data set, el exponente de ponderación m, el número de clusters, la medida de distancia a utilizar y el error de tolerancia; su escalabilidad es dada por n muestras para c clusters pudiendo ser estos grandes conjuntos de datos. El uso permite determinar funciones de membresía para conjuntos difusos, donde estas funciones de membresía son curvas particulares descritas por los vectores $[u_r]$ conocidos como vectores de partición; además los clusters resultados son conjuntos difusos que permiten dar un análisis descriptivo para ser denotados mediante operadores lingüísticos, además permiten establecer las medidas de distancias como no euclidianas; también los centroides producto del algoritmo pueden ser interpretados como prototipos difusos, como se revisó anteriormente.

El algoritmo de ***Possibilistic C-Means*** es muy parecido al de FCM pero un parámetro es adicionado η el cual es un vector de números positivos definidos para cada clúster, pues en la función de optimización similar a la de FCM es agregado un término que le suma un grado de robustez al algoritmo, pero también aumenta el la dificultad en implementación y uso de este, debido a la cantidad de parámetros que quedan a criterio del usuario.

2.6. Comparación de medidas de validación

A continuación se realizará un análisis comparativo para los métodos de clustering de acuerdo a resultados en trabajos anteriores como base para el contrate entre estos. Wang y Zhang [53] llevaron a cabo un experimento con las medidas de validación antes mencionadas en ocho data sets públicos y ocho data sets creados artificialmente de los cuales sus categorías eran previamente conocidas, cuyos resultados se van a expresar a continuación.

V_{PC} y V_{PE} son medidas que tienden a identificar incorrectamente el número de clusters c dado a que tienden monótonamente en función de este parámetro. V_{MPC}, V_{KYI} y V_P son medidas que tienden a identificar correctamente el número de clusters c con un relativamente bajo rango de fallo, reconociendo fallidamente un número c óptimo en un rango de 2 a 3 data sets; esto cuando los índices dependen de los valores de membresía netamente.

En los 18 índices de validación que involucran los valores de membresía y el data set se encontraron los siguientes resultados; V_{FS} tiene un rango de fallo considerable ya que en 5 de los 16 data sets estimó un número de clusters c incorrecto. V_{XB}, V_K y V_T poseen resultados similares identificando 4 data sets con un número c

incorrecto e incluso V_{XB} identificó mal un c en otro data set adicional a los 4 que tiene en común con las otras medidas. V_{SC} identificó correctamente la mayoría de los c para los data sets a excepción de 3. V_{FHV}, V_{APD} y V_{PD} identificaron coincidentemente mal el número c, pero el índice V_{APD} identificó mal 2 número de clusters para data sets públicos teniendo un bajo índice de falla. V_{PCAES} identificó en el experimento 13 de los 16 data sets con un número c óptimo correcto. $V_{S_{VI}}$ reconoció erróneamente 7 de los 16 data sets con un c incorrecto. V_{CWB} identificó correctamente 14 de los 16 data sets, solo fallando en 2 de estos. V_{WSJ} falló en el reconocimiento de 5 de los 16 data sets siendo también un rango de error considerable. V_{OS} identificó falsamente un c óptimo en 6 data sets, también un error considerable. V_{PBMF} identificó solamente un data set con un c incorrecto, siendo el índice de validación con el menor error dentro del experimento. Continuando, V_{SCG} identificó a 4 data sets con un número c óptimo incorrecto. V_D identificó 7 de los 16 data sets con números de clusters c incorrectos, con casi un 50% de error dentro del experimento. Terminando con los índices V_{GD} y $V_{cond(H)}$ los cuales identificaron erróneamente un c para 2 data sets. Wang y Zheng también expresan que en su observación los índices de validación que dependen de los valores de membresía y del data set, además de

agregar robustez al análisis y validación, son muy sensibles al ruido provisto por el data set, como el caso de los índices: V_{PCAES}, V_{CWB}, V_{WSJ}, V_{PBMF}, V_{GD} y $V_{cond(H)}$ pues el ruido se presenta por datos que no tienen el potencial suficiente para que el valor de membresía le permita definirse dentro de un clúster bien identificado.

2.7. Conclusiones

Las técnicas de clustering permiten identificar estructuras de sub-agrupaciones dentro de un conjunto de datos de acuerdo a medidas de afinidad o similitud; entre las técnicas de clustering tenemos K-means, el método de propagación por afinidad, el clustering aglomerativo, FCM y Possibilistic C-means; en los cuales se realizó una revisión de la literatura pertinente y una comparación basada en parámetros de criterio orientados a esta investigación. Estos criterios son: parámetros utilizados, escalabilidad y casos de uso; el número de parámetros del que depende el algoritmo permite establecer un tamaño estimado del software para su posterior implementación, la escalabilidad en el contexto de sub-agrupaciones o clusters permite definir un diseño de pruebas y experimentos; mientras que los casos de uso

soportan el diseño de los componentes de software de acuerdo a cuál es la tarea de dicho componente.

El prototipado difuso es una técnica que permite extraer conocimiento de un grupo de datos mediante un dato que caracterice la muestra, esto mediante medidas de semejanza y disimilitud que permitan establecer su grado de tipicidad; pero esta técnica de prototipado permite establecer en un contexto de lógica difusa no solo la afinidad de un dato a un prototipo exclusivo, sino más bien por el grado de pertenencia a cada cluster. Este proceso de prototipado permitirá en la etapa de diseño, junto con el algoritmo de clustering, definir formas de categorización no supervisada y mediante técnicas de defusificación la clasificación de nuevos datos y sus características.

Debido que el diseño del modelo está estrechamente relacionado con las técnicas de clustering, existe una importancia en la validación de estos procesos, ya que una de las características es que no existe una categorización conocida previamente, en el contexto de rendimiento o comportamiento académico de estudiantes. El objetivo de la validación de clustering difusos es encontrar un número de clusters c óptimo, se llevó a cabo una revisión de estas medidas de validación y finalmente se realizó una

comparación explícita sobre los índices de validación, en los que la literatura describe algunos de estos criterios de comparación como: comportamiento monótono, sensibilidad a parámetros como el exponente de ponderación y sensibilidad al mismo data set; en un experimento exhaustivo se llegó a la conclusión de que ninguna de estas medidas reconoce correctamente un número óptimo de clusters, pues la complejidad y robustez del índice influyen en la precisión de la optimización pero al final depende en gran medida de la data.

CAPITULO 3

3. ANÁLISIS Y DISEÑO DEL MODELO DE CLUSTERIZADO MULTINIVEL

3.1. Análisis

3.1.1. Análisis de requerimientos funcionales

En este punto se detallarán las operaciones y funciones principales del prototipo a desarrollar, así como los requerimientos específicos que servirán como base para el desarrollo del mismo.

Operaciones:

1. Pre-procesar los datos para realizar el cálculo de la estimación

2. Realizar la selección del vector de atributos para la estimación

3. Calcular la estimación del riesgo de reprobación para un estudiante, en un semestre dado

Funciones:

- Consultar y registrar estimaciones
- Consultar y registrar datos de estudiantes
- Consultar y registrar datos de semestres
- Consultar y registrar datos de materias
- Búsqueda de estimaciones por estudiante
- Generar estimaciones como archivos

Características de los usuarios

Los usuarios no interactúan directamente con el software diseñado puesto que este no tendrá un componente de interfaz gráfica. Las características de los usuarios son legadas por el sistema de Consejerías, del que se propone sea parte este módulo, como profesores y estudiantes usuarios.

Asunciones y dependencias

El sistema dependerá de una base de datos que contenga información actualizada de estudiantes y términos académicos, este proceso de actualización no lo realizará el componente o prototipo propuesto.

Requerimientos específicos

Requerimientos de desempeño. El módulo a desarrollar debe poder ejecutarse según las solicitudes de usuarios y el volumen de acceso de estos, de acuerdo a las especificaciones del sistema en el que se integrará; el servicio de base de datos y la provisión de webservices se deben mantener inalterables para la correcta función del sistema. El 95% de las transacciones hechas por el componente y provistas mediante las interfaces a otros sistemas o módulos se realizarán en menos de 1 segundo. El consumo de datos provistos debe mantenerse estable durante un 97% del tiempo en el que el módulo esté integrado y habilitado al sistema.

Requerimientos lógicos de Base de Datos. Los datos usados por el componente serán almacenados en un servidor de bases de datos al que el componente accede para consumir y almacenar los datos de entrenamiento calculados y sus estimaciones; en esta base de datos se almacenarán los registros correspondientes a las entidades de estudiantes y semestres, los datos clasificados para

entrenamiento y las estimaciones calculadas. El almacenamiento de los datos generados se realizará siempre que existe un nuevo proceso de cálculos, este proceso debe ser realizado en un promedio de 2 veces cada 4 meses.

Limitaciones de Diseño. Los componentes propuestos están diseñados para trabajar en un ambiente web y su interacción mediante un navegador; El diseño de este módulo se verá sujeto a la independencia de los componentes descritos en el transcurso de este capítulo.

Cumplimiento de Estándares. El componente no debe de permitir la eliminación o edición de datos calculados una vez que estos estén registrados en la base, así también la manipulación directa de los datos no debe ser permitida a excepción de la manipulación por usuarios con permisos de edición previamente especificados.

Especificación de requerimientos funcionales

Componente de Pre-procesamiento

Tabla 3. Especificación de requerimientos funcionales RF-1

RF-1	Consumo de datos de Estudiantes
Versión	1.0

Tipo	Nuevo
Dependencias	
Descripción	El componente a implementar debe poder consumir los datos de estudiantes como datos académicos y socioeconómicos que serán usados en el proceso de estimación para determinar el riesgo de reprobación para estudiantes.
Importancia	5
Urgencia	3
Comentarios	
Forma de validación	Simulación

Tabla 4. Especificación de requerimientos funcionales RF-2

RF-2	Consumo de datos de Materias
Versión	1.0
Tipo	Nuevo
Dependencias	
Descripción	El componente a implementar debe poder consumir los datos de materias asociados a estudiantes para realizar la estimación de basado en las calificaciones de los estudiantes y poder determinar atributos como la rigurosidad y dificultad relacionados con una materia.

Importancia	5
Urgencia	3
Comentarios	
Forma de validación	Simulación

Tabla 5. Especificación de requerimientos funcionales RF-3

RF-3	Cálculo de atributos de estudiantes
Versión	1.0
Tipo	Nuevo
Dependencias	RF-1
Descripción	El componente a implementar debe poder calcular los atributos relacionados a un estudiante como el Grade Point Average (GPA), el Promedio Académico, número de semestres que ha cursado, rendimiento, número de materias tomadas, número de materias aprobadas y número de materias reprobadas.
Importancia	5
Urgencia	3
Comentarios	
Forma de	Simulación

validación	

Tabla 6. Especificación de requerimientos funcionales RF-4

RF-4	Cálculo de atributos de semestres
Versión	1.0
Tipo	Nuevo
Dependencias	RF-1 y RF-2
Descripción	El componente a implementar debe poder calcular los atributos relacionados de semestres relacionados con estudiantes como: la dificultad, rigurosidad y skewness total en función de las materias que pertenecen al semestre; así como el número de semestre que fue cursado por un estudiante, el total de materias que pertenecen al semestre y el ratio de reprobación.
Importancia	5
Urgencia	3
Comentarios	
Forma de validación	Simulación

Componente de Clustering

Tabla 7. Especificación de requerimientos funcionales RF-5

RF-5	Clustering de estudiantes
Versión	1.0
Tipo	Nuevo
Dependencias	RF-3
Descripción	El componente realizará un clustering de estudiantes basado en el vector de atributos especificado para obtener un data set entrenado y usarlo en clasificación.
Importancia	5
Urgencia	4
Comentarios	
Forma de validación	Simulación

Tabla 8. Especificación de requerimientos funcionales RF-6

RF-6	Clustering de semestres
Versión	1.0
Tipo	Nuevo

Dependencias	RF-4
Descripción	El componente realizará un clustering de semestres basado en el vector de atributos especificado para obtener un data set entrenado y usarlo en clasificación.
Importancia	5
Urgencia	4
Comentarios	
Forma de validación	Simulación

Tabla 9. Especificación de requerimientos funcionales RF-7

RF-7	Validación de Clustering
Versión	1.0
Tipo	Nuevo
Dependencias	RF-5 y RF-6
Descripción	El componente realizará una validación de índices de clustering para validar los parámetros de entrada del algoritmo de FCM.
Importancia	5
Urgencia	2

Comentarios	
Forma de validación	Simulación

Componente de Clasificación

Tabla 10. Especificación de requerimientos funcionales RF-8

RF-8	Clasificación basada en datos entrenados
Versión	1.0
Tipo	Nuevo
Dependencias	RF-5, RF-6 y RF-7
Descripción	El componente realizará una clasificación mediante técnicas de inteligencia artificial como Support Vector Classifier o defusificación para realizar la estimación en base a la semejanza de entidades.
Importancia	5
Urgencia	3
Comentarios	
Forma de validación	Simulación

Tabla 11. Especificación de requerimientos funcionales RF-9

RF-9	Estimación de riesgo
Versión	1.0
Tipo	Nuevo
Dependencias	RF-8
Descripción	La estimación de riesgo debe de realizarse con base en técnicas estadísticas para lo cual se realizará un análisis descriptivo de las razones de aprobación y reprobación y poder generalizar estas razones a nivel de datos preclasificados.
Importancia	5
Urgencia	4
Comentarios	
Forma de validación	Simulación

3.1.2. Análisis de requerimientos no funcionales

Confiabilidad. Al ser un componente propuesto para un sistema académico, este debe proporcionar el manejo de errores en caso de que ocurra y no llegar a deteriorar o desestabilizar el sistema, ni mucho menos conducir al sistema al colapso; puesto que la información podría perderse durante el proceso de registro de datos. Para evitar inconvenientes como la incomodidad de

usuarios, el colapso, degradación o inestabilidad del sistema, el componente debe tener un alto grado de confiabilidad.

Disponibilidad. El módulo debe proveer datos, mediante sus interfaces un 98% del tiempo en que esté ejecutándose; además, el módulo debe realizar el cálculo para estimaciones un 99% de las veces de forma exitosa y esta operación debe estar disponible al culminar cada semestre.

Seguridad. La integridad de la información de los estudiantes y de los cálculos para estimación de riesgo deben mantenerse restringidos por los roles establecidos por el sistema. Los datos registrados no deben poder ser manipulados sino mediante los procesos correspondientes.

Mantenibilidad. Para facilidad del mantenimiento, el módulo desarrollado implementará patrones de diseño convencionales y las estrategias de inteligencia artificial utilizadas serán detalladas y explicadas en el capítulo de implementación. Los componentes proporcionan desacoplamiento entre las funcionalidades del sistema para cumplir con escalabilidad, y poder adaptarse al mantenimiento presente en el tiempo.

Portabilidad. El desarrollo basado en componente permite al software desarrollado poder ser integrado de manera modular en

un sistema que pueda soportar sus interfaces. El módulo desarrollado también puede ser ejecutado en cualquier sistema operativo con soporte para Python integrando las librerías de terceros usadas.

3.2. Casos de uso

En este punto se describen los casos de usos planteados para el módulo de estimación de riesgo propuesto y detalle de los actores mediante diagramas UML; en estos diagramas se especificarán los casos de usos presentados en los diferentes subcomponentes.

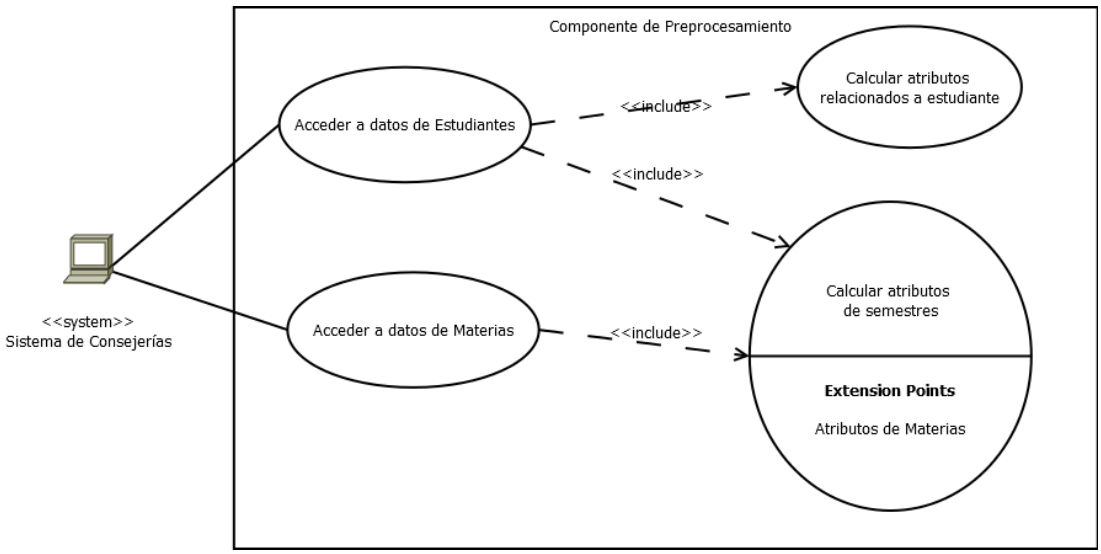


Figura 3.1. Diagrama de casos de uso para el componente de pre-procesamiento

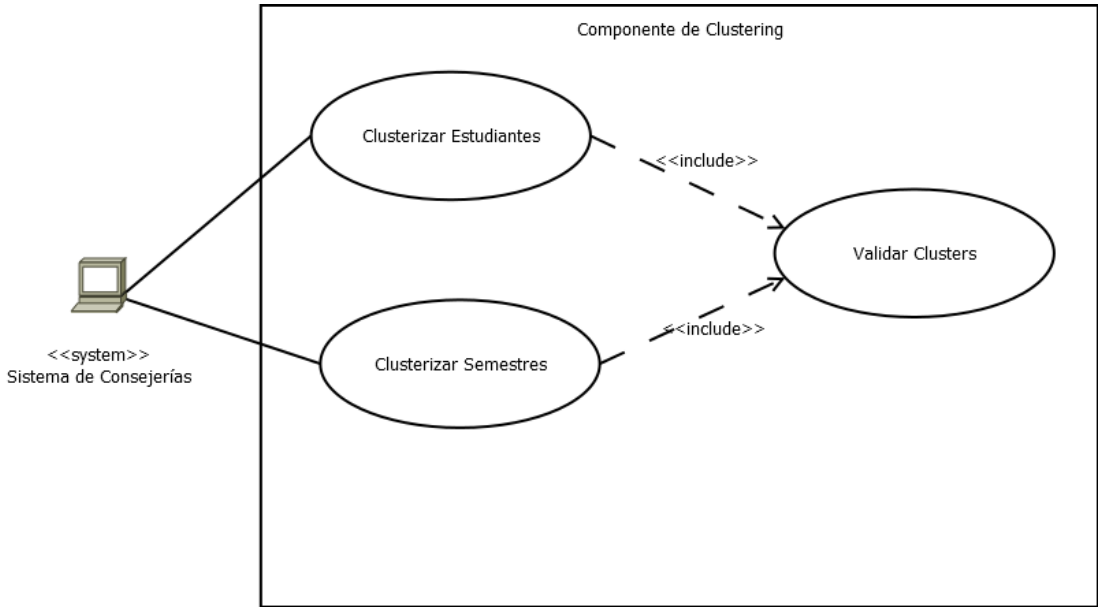


Figura 3.2. Diagrama de casos de uso del componente de clustering

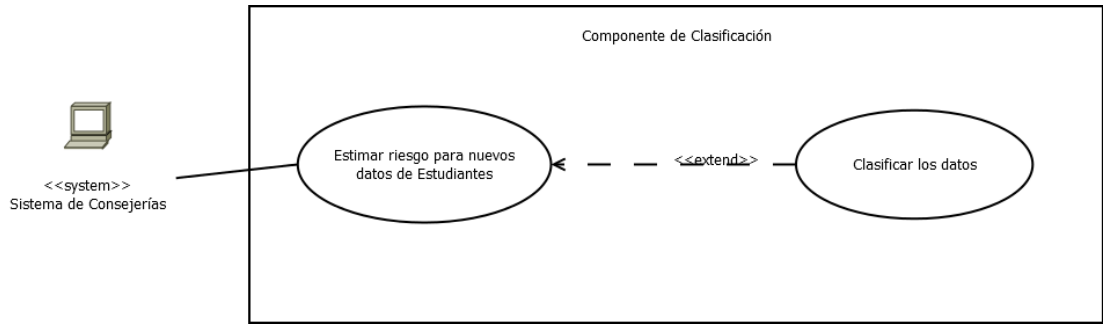


Figura 3.3. Diagrama de casos de uso del componente de clasificación

3.3. Diseño

El diseño del módulo de estimación de riesgo, está basado en un cluster de dos niveles en combinación de ejes que dependen de las características del semestre y las características del estudiante; este cluster de dos niveles especifica dos tipos de similitudes: entre semestres y entre estudiantes; las que se identificará como ejes, lo que nos da como resultado los siguientes niveles de similitud especificados a continuación:

Eje de Similitud Entre Estudiantes

Primer Nivel: Clustering de Estudiantes

Clustering Basado en Factores resultado del EFA

- Factor 1: Basic Training Factor
- Factor 2: Advanced CS Topics Factor
- Factor 3: Client Interaction Factor
- Factor 4: Programming Factor
- Factor 5: No Related to CS Factor

Eje de Similitud Entre Semestres

Segundo Nivel: Clustering de Semestres

Clustering basado en medidas de:

- dificultad
- rigurosidad
- distribución de notas
- número de materias tomadas
- semestre actual

3.4. Componentes

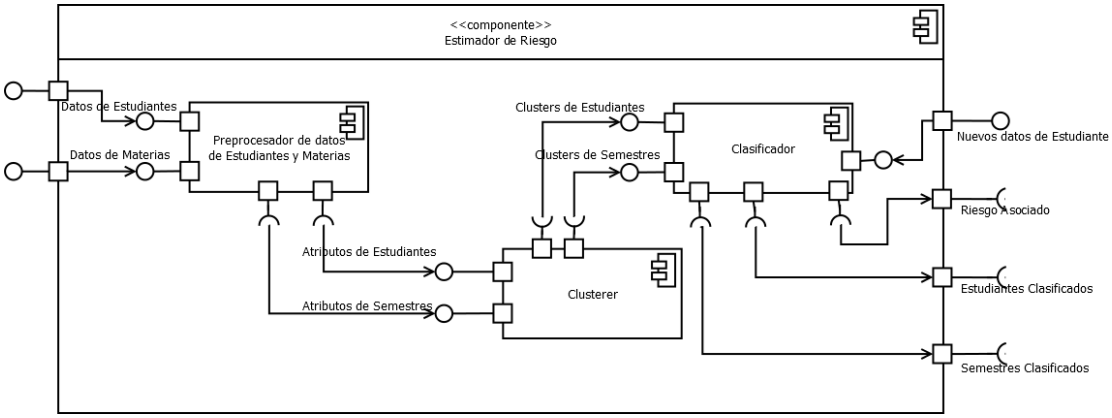


Figura 3.4. Diagrama de componentes del módulo de estimación de riesgo

Los componentes que forman el módulo de estimación de riesgo son: pre-procesamiento, clustering y clasificación. El componente de pre-procesamiento es el encargado de calcular atributos para cada elemento del data set, y hacer una selección de atributos que van a ser usados en el proceso de clustering. El componente de clustering es el responsable del proceso de agrupación de semestres y estudiantes, y también es el encargado de validar los clusters. El componente de clasificación es el encardo de usar el

data set clusterizado, el cual será utilizado como dato de entrenamiento para un clasificador no supervisado, y asociar el riesgo a cada clúster, esto es, la estimación se dará por la generalización del riesgo asociado a un clúster en el que el nuevo dato es clasificado.

3.4.1. Componente de pre-procesamiento

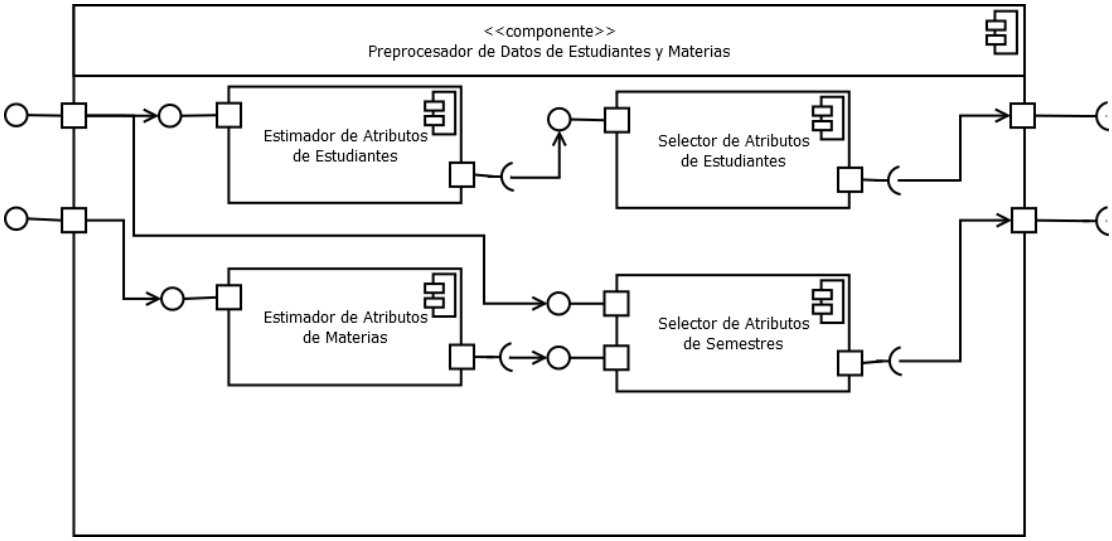


Figura 3.5. Elementos internos del componente pre-procesamiento

Este componente utiliza los datos de estudiantes y materias para realizar el cálculo de los atributos asociados a estudiantes, como medidas de habilidades adquiridas y promedio académico, y también calcula los atributos asociados a materias, como las

medidas de dificultad y rigurosidad; estos atributos son seleccionados para ser usados en el componente de clustering.

3.4.2. Componente de clustering

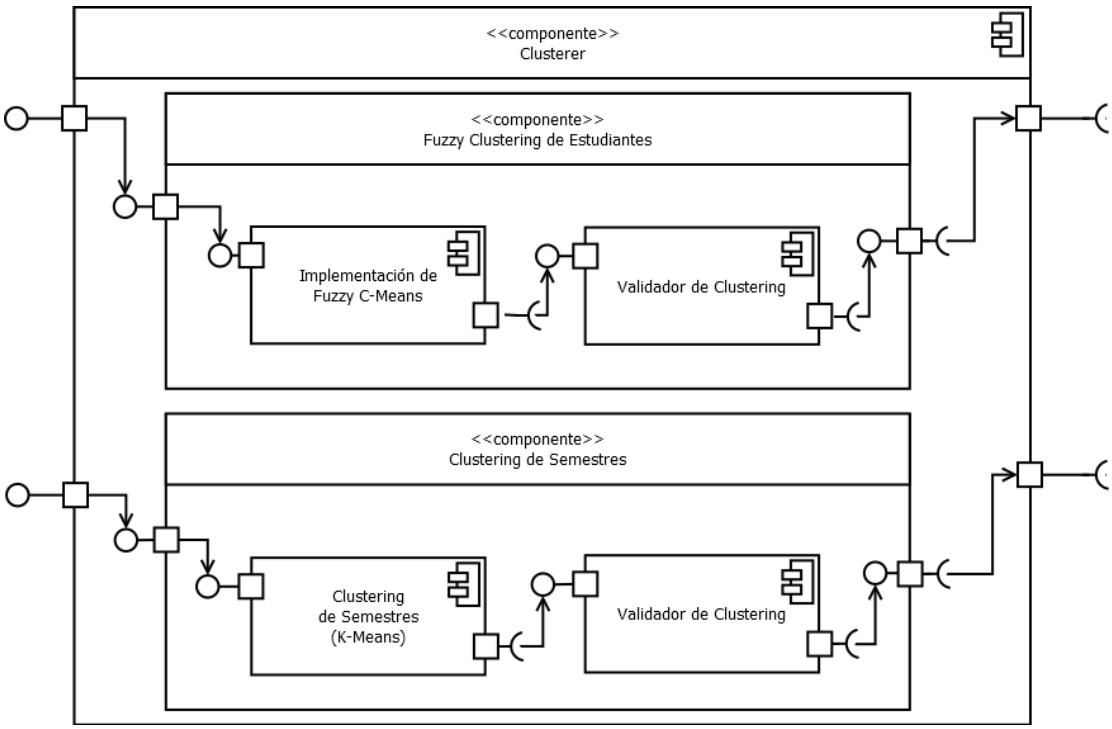


Figura 3.6. Elementos internos del componente de clustering

Este componente es el encardo del proceso de clustering para estudiantes y semestres; los clusters resultantes son validados para obtener un número de clusters apropiado.

3.4.3. Componente de clasificación

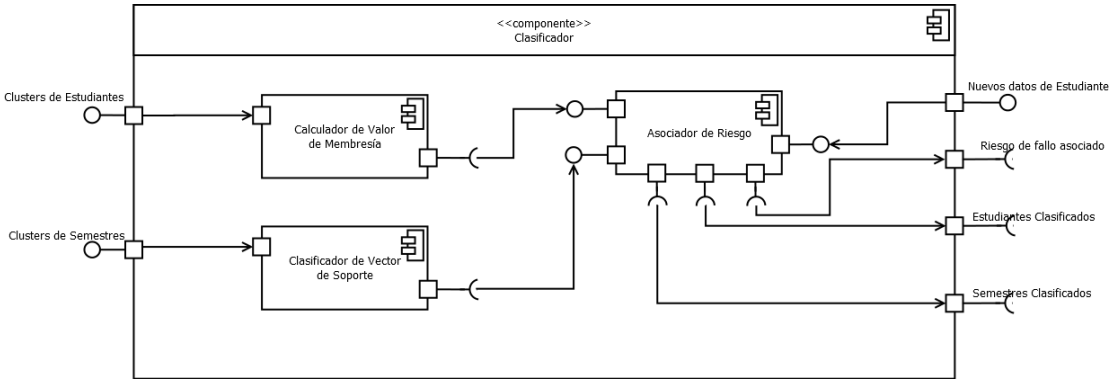


Figura 3.7. Elementos internos del componente de clasificación

El Componente de Clasificación interpreta los clusters de estudiantes y semestres como categorías en las que pueden ser clasificados los nuevos datos; estos clusters de estudiantes y semestres son usados como datos de entrenamiento para el clasificador no supervisado de semestres y la función de membresía para estudiantes; además este componente asocia el riesgo de fallo en materias a cada clúster de estudiantes y semestres, permitiendo a los nuevos pares de estudiantes-semestres, asociarlos con un valor de riesgo al ser clasificados dentro de los clusters.

3.5. Diseño de experimentos y pruebas

Experimento de Validación de Clusters. Se realizaron varios experimentos para determinar que índice de validación permite

identificar adecuadamente los parámetros m y c para el algoritmo de FCM; para esto, se usaron tres diferentes índices de validación: un índice de validación convencional, un índice que dependa de medidas de membresía, y un índice que dependa de medidas de membresía y el data set. Luego se revisaron los clusters creados, con gráficos de radar para verificar si existen diferencias notables con respecto al número de clusters sugeridos por los índices de validación.

Experimento de Verificación de Estimación de Riesgo. Mediante el Score de Brier [55] se determinó si la estimación del riesgo realizada tiene un porcentaje de certeza aceptable, para esto se utilizó los datos de los semestres y estudiantes desde el año 1978 hasta el 2012, que constituyeron los datos de entrenamiento, asociando a los clusters un riesgo de fallo; luego, se clasificaron datos de estudiantes y semestres del 2013 como nuevos datos; finalmente se verificará el riesgo estimado con el porcentaje de reprobación.

Diseño de Pruebas Unitarias. Se realizaron pruebas unitarias basadas en la funcionalidad de los componentes implementados. La documentación de las pruebas unitarias se realizó utilizando el estándar 829-2008 de la IEEE, para plan de pruebas [54].

Diseño de Pruebas de Rendimiento. Las pruebas de rendimiento se llevaron a cabo sobre partes del código que se consideran de vital importancia para el funcionamiento del componente, estas son: el proceso de clustering, el ajuste del clasificador, la estimación del grado de membresía para nuevos datos, y la asociación de riesgo a cada clúster.

CAPITULO 4

4. IMPLEMENTACIÓN

4.1. Diseño de experimentos y pruebas

En este capítulo se detallarán las partes fundamentales del diseño e implementación de los componentes de pre-procesamiento, clustering y clasificación, en forma de pseudocódigo, desglosando el funcionamiento y la estructura del mismo. El prototipo fue desarrollado en Python, un lenguaje de alto nivel que permite trabajar de forma flexible las expresiones matemáticas complejas y operaciones con estructuras abstractas como conjuntos, que son la base de este modelo de predicción.

4.1.1. Componente de pre-procesamiento

Este componente realiza el cálculo de los atributos relacionados con el rendimiento académico por estudiante, y también los

atributos relacionados con la carga académica por semestre; a continuación se detalla el cálculo de los atributos relacionados con rendimiento académico utilizando la estructura “estudiante”.

Tabla 12. Pseudocódigo para el cálculo del rendimiento de estudiantes

```
/* Cálculo del rendimiento de estudiantes*/
LIST studentsTrainData
FOR each student IN computer_science
     SET i to 1
     LIST studentFeatures
     WHILE i ≤ 5
           /*get skill measure in factor F(i)*/
           SET skillMeasure(i) to AVG scores from (courses ∈ F(i)) ∧
           (courses ∈ student academic history)
           APPEND skillMeasure(i) to studentFeatures
           INCREMENT i
     END WHILE
     APPEND studentFeatures to studentsTrainData
END FOR
```

Los atributos relacionados con el rendimiento académico dependen de la historia académica, su cálculo se basa en las calificaciones de los cursos tomados, y además estos cursos deben estar dentro de los cinco factores mencionados en el punto 2.4.5, que describen las habilidades desarrolladas.

Los atributos relacionados con la carga académica, están dados por la combinación única de materias que alguna vez ha tomado un estudiante durante su historia académica, y dependiendo de

esta combinación de materias, se calcula la dificultad, rigurosidad y skewness por semestre; además, también se obtiene el número de materias en ese semestre y el nivel relativo promedio en el que están los estudiantes que toman dicha combinación de materias.

Tabla 13. Pseudocódigo para el cálculo de carga semestral

```
LIST semestersTrainData
FOR each semester IN general academic history
     SET semesterAlpha to 0
     SET semesterBeta to 0
     SET semesterSkewness to 0
     SET semesterCoursesN to LENGTH semester
     SET semesterLevelMean to CALL levelMean(semester)
     FOR each course IN semester
            SET semesterAlpha to semesterAlpha + CALL getDifficulty(course)
            SET semesterBeta to semesterBeta + CALL getStringency(course)
            SET semesterSkewness to semesterSkewness + CALL getSkewness(course)
     END FOR
     SET semesterFeatures to {semesterAlpha, semesterBeta, semesterSkewness,
     semesterCoursesN, semesterLevelMean}
     APPEND semestersFeatures to semestersTrainData
END FOR
```

Tabla 14. Pseudocódigo para el cálculo del nivel relativo al estudiante

```
SET relativeLevel to 0
FOR each student IN computer science
     IF student HAS semester
            SET relativeLevel to relativeLevel + student
     academicHistory[semester] level
     END IF
END FOR
SET levelMean to relativeLevel / COUNT student HAS semester
```

4.1.2. Componente de clustering

Este componente usa la implementación de los algoritmos de clustering para construir los prototipos difusos y las categorías que alimentarán al componente de clasificación y asociación; el clustering de estudiantes y semestres es realizado por separado, ya que son estructuras distintas, creando dos niveles de clusters; es decir, se asume que existen varios tipos de semestres los cuales están relacionados con varios tipos de estudiantes. El clustering de estudiantes es realizado con el algoritmo FCM mientras que el clustering de semestres es realizado con el algoritmo K-Means. El siguiente pseudocódigo detalla el uso de los algoritmos para la construcción de los prototipos y clusters de estudiantes y su validación respectiva para determinar el número de categorías.

Tabla 15. Pseudocódigo para el proceso de clustering en estudiantes

```
SET ε to 1·10^-10
LIST vData
FOR (c,m) IN (2..16]×(1..5]
        SET p_r, U to CALL fcm(studentsTrainData, c, m, ε)
        STORE p_r
        SET V_MPC to 1 - c·(1 - SUM (u_ir)^2 from U / COUNT studentsTrainData) / (c -
1)
        APPEND (V_MPC, m, c) to vData
END FOR
STORE vData
```

Tabla 16. Pseudocódigo para la validación del clustering de estudiantes

```
SET vMax to 0
SET mMax to 0
SET cMax to 0
FOR each (V_MPC, m, c) IN vData
         IF vMax ≤ V_MPC
                    SET vMax to V_MPC
                    SET mMax to m
                    SET cMAx to c
     END IF
END FOR
STORE (vMax, mMax, cMax)
```

Tabla 17. Pseudocódigo para el proceso de clustering en semestres

```
LIST vData
FOR c IN [2..16]
         SET kmeans to CALL KMeans(c)
         CALL kmeans.fit(studentsTrainData)
         SET w_r to kmeans.cluster_centers
         STORE w_r
         SET cIDs to kmeans.compute()
     STORE cIDs
         SET V_DUNN to CALL dunnIndex(studentsTrainData, kmeans.clusters, c)
         APPEND (V_DUNN, c) to vData
END FOR
STORE vData
```

Tabla 18. Pseudocódigo para el cálculo del índice de Dunn

```
SET maxΔ to MAX pairwiseDistance(studentsTrainData)
FOR i IN [1..c]
         FOR j IN [1..c]
                    IF j≠i
                              SET V_DUNN to pairwiseDistance(kmeans.clusters[i],
kmeans.clusters[j]) / maxΔ
```

```
                    END IF
     END FOR
END FOR
```

Tabla 19. Pseudocódigo para la validación del clustering de semestres

```
SET vMax to 0
SET cMax to 0
FOR each (V_DUNN, c) IN vData
         IF vMax ≤ V_MPC
                    SET vMax to V_DUNN
                    SET cMAx to c
     END IF
END FOR
STORE (vMax, cMax)
```

4.1.3. Componente de clasificación

Este componente realiza la clasificación de los nuevos datos, y extrae las variables descriptivas de los clusters provistos por el componente de clustering. Para los estudiantes, la extracción de las variables se realiza con una defusificación mediante la membresía máxima, el máximo de la función de membresía también es usado para la clasificación de nuevos datos. Para el caso de las categorías de semestres, se utiliza una máquina de soporte de vector con un kernel RBF, con una penalidad de 0.1 y un γ igual a 0.0 entrenada con las categorías de semestres.

Tabla 20. Pseudocódigo para la clasificación por membresía máxima de estudiantes

```
/* Clasificación por membresía máxima de estudiantes, newData es el dato a
clasificar */
LIST defuzzified
SET u_ir to CALL fcm.membership(newData, p_r, cMax, mMax)
SET mm to MAX u_ir
APPEND mm to defuzzified
```

Tabla 21. Pseudocódigo para la clasificación por máquina de soporte de semestres

```
/* Clasificación por máquina de soporte de semestres, newData es el dato a
clasificar */
SET C to 0.1
SET γ to 0.0
SET clf to svm(RBF_KERNEL, C, γ)
CALL clf.fit(w_r, w_r.indexes)
SET cID to CALL clf.predict(newData)
```

Tabla 22. Pseudocódigo para la extracción de variables descriptivas

```
MATRIX reprovingCases
MATRIX takenCases
MATRIX reprovingRates
SET reprovingCases to [0]_MxN
SET takenCases to [0]_MxN
SET reprovingRates to [0]_MxN
FOR each (student, mm, semester, cID) IN (studentsTrainData,
defuzzified)×(semestersTrainData, cIDs)
        IF semester.isReproved
                SET reprovingCases[mm, cID] to reprovingCases[mm, cID] + 1
        SET takenCases[mm, cID] to takenCases[mm, cID] + 1
        END IF
END FOR
FOR each mm IN COUNT takenCases.rows
        FOR each cID IN COUNT takenCases.column
```

```
                    SET reprovingRates[mm, cID] to reprovingCases[mm, cID] /
takenCases[mm, cID]
     END FOR
END FOR
STORE reprovingRates
```

4.1.4. Librerías y componentes de terceros

A continuación se describen las librerías y componentes de terceros usados en la implementación del módulo.

- ***Numpy***. Es un paquete para computación científica con Python, permite manejar objetos de n-dimensiones y realizar operaciones basadas en álgebra lineal [56]. El manejo de grandes conjuntos con este paquete permite hacer búsquedas y operaciones básicas sobre vectores, esto es esencial para el pre-procesamiento de datos.

- ***Pandas***. Librería que provee manejo de estructuras de datos, como Series y DataFrames, mediante herramientas de análisis de datos para Python [57]. Esta librería permite abstraer mediante DataFrames las estructuras de estudiantes, materias y semestres, haciendo más fácil su manejo

- ***SciKitLearn***. Paquete para aprendizaje de máquina en Python, provee herramientas de minería de datos como clasificación, regresión, clustering, reducción de dimensiones, entre otros

[58]. En este paquete se encuentran las implementaciones usadas para k-Means y de máquina de vector de soporte para clasificación no supervisada.

- ***SciKitFuzzy*** [59]. Librería que posee varias implementaciones de algoritmos de lógica difusa, entre estos algoritmos se encuentra la implementación de FCM, usada para la construcción de prototipos difusos.

4.2. Costos asociados y plan de implementación

Los principales costos asociados en esta investigación y desarrollo del módulo son el talento humano y el uso de hardware. La investigación y revisión de literatura tomó cinco meses y la implementación de un prototipo, basada en el uso de un desarrollador, tomó tres meses. El diseño y ejecución de las pruebas y validaciones respectivas tomó un mes. El desarrollo del software tomó en total nueve meses, considerando el uso de un solo recurso para su investigación, desarrollo y pruebas. El hardware utilizado fue una computadora personal con un procesador Intel Xeon de ocho núcleos y ocho gigabytes de memoria RAM.

CAPITULO 5

5. RESULTADOS EXPERIMENTALES Y PRUEBAS

5.1. Pruebas unitarias

Las pruebas unitarias fueron ejecutadas utilizando el diseño planteado en la fase de diseño. A continuación se detallan los resultados de estas pruebas.

5.1.1. Casos de prueba

Características que no serán probadas. No se realizarán pruebas sobre el código de terceros, que son parte de los paquetes y librerías: Numpy, Pandas, ScikitLearn y ScikitFuzzy, y que aporten al desarrollo del módulo y sus componentes, se asume que estas pasaron por un proceso de pruebas, antes de ser liberadas de forma pública.

- **Especificación del diseño de pruebas para cálculo de atributos de un estudiante**

  Especificación del diseño de prueba. TDS-01, cálculo de atributos para un estudiante.

  Características a ser probadas. Validar tipos de datos ingresados y datos dentro del rango permitido.

  Identificación de pruebas:

  TC-001: Se intenta calcular los atributos para un estudiante ingresando una matrícula que no es válida.

  TC-002: Se intenta ingresar datos no numéricos como matrícula de un estudiante.

  TC-003: Se intenta calcular los atributos para un estudiante sin historia académica.

  TC-004: Se intenta calcular los atributos para un estudiante con historia académica.

- Caso de prueba TC-001

  Especificación del Identificador del caso de prueba. TC-001, Se intenta calcular los atributos para un estudiante ingresando una matrícula que no es válida.

Ítems de Prueba. Se prueba el ingreso de una matrícula válida. Si no se ingresa una matrícula válida de un estudiante no se debe permitir el cálculo de los atributos.

Especificación de entradas:

matricula=301625248

Especificación de salidas:

factor1=NaN

factor2=NaN

factor3=NaN

factor4=NaN

factor5=NaN

Dependencia de otros casos de prueba. Ninguna

- Caso de prueba TC-002

  Especificación del Identificador del caso de prueba. TC-002, Se intenta ingresar datos no numéricos como matrícula de un estudiante.

  Ítems de Prueba. Se probará el ingreso de una matrícula válida. Si no se ingresa una matrícula válida de un estudiante no se debe permitir el cálculo de los atributos.

  Especificación de entradas:

  matrícula=aaaaaaaaa

Especificación de salidas:

factor1=NaN

factor2=NaN

factor3=NaN

factor4=NaN

factor5=NaN

Dependencia de otros casos de prueba. Ninguna

- Caso de prueba TC-003

  Especificación del Identificador del caso de prueba. TC-003, Se intenta calcular los atributos para un estudiante sin historia académica.

  Ítems de Prueba. Se prueba el cálculo de atributos para un estudiante sin historia académica (estudiante que no ha cursado algún semestre pero ha ingresado a la institución). Si un estudiante no posee historia académica los atributos calculados son igual a cero.

  Especificación de entradas:

  matricula=201314848

  Especificación de salidas:

  factor1=0.0

  factor2=0.0

factor3=0.0

factor4=0.0

factor5=0.0

- Caso de prueba TC-004

  Especificación del Identificador del caso de prueba. TC-004, Se intenta calcular los atributos para un estudiante con historia académica.

  Ítems de Prueba. Se prueba el cálculo de atributos para un estudiante con historia académica. Si un estudiante posee historia académica los atributos calculados serán mayores o iguales a cero.

  Especificación de entradas:

  matrícula=199803404

  Especificación de salidas:

  factor1=7.46875

  factor2=7.241666667

  factor3=7.5375

  factor4=6.25

  factor5=7.525

  Dependencia de otros casos de prueba. Ninguna

- **Especificación de Casos de Pruebas para cálculo de atributos para un semestre**

  Especificación del diseño de prueba. TDS-02, cálculo de atributos para un semestre.

  Enfoque específico. Se valida que la lista de códigos de materias ingresadas sea ingresada ya que es un campo obligatorio.

  Características a ser probadas. Validar tipos de datos ingresados y datos obligatorios.

  Identificación de pruebas:

  TC-005: Se intenta calcular los atributos para un semestre en el que se toma solo una materia y esta no tiene un código válido.

  TC-006: Se intenta calcular los atributos para un semestre en el que una de las materias tomadas no tiene un código válido.

  TC-007: Se intenta calcular los atributos para un semestre en el que se toma solo una materia y esta tiene un código válido.

  TC-008: Se intenta calcular los atributos para un semestre en el que todas las materias tomadas tienen un código válido.

TC-009: Se intenta calcular los atributos para un semestre con una lista de códigos de materias vacía.

- Caso de prueba TC-005

  Especificación del Identificador del caso de prueba. TC-005, Se intenta calcular los atributos para un semestre en el que se toma solo una materia y esta no tiene un código válido.

  Ítems de Prueba. Se prueba el ingreso de la lista de códigos de materias tomadas en un semestre. Si ninguno de los códigos de las materias tomadas en el semestre son códigos válidos no se le asignará valores a los atributos del semestre.

  Especificación de entradas:

  materias=XXXXZZZZ

  Especificación de salidas:

  total_difficulty=NaN

  total_stringency=NaN

  total_skewness=NaN

  classes_num=NaN

  level_mean=NaN

  Dependencia de otros casos de prueba. Ninguna

- Caso de prueba TC-006

Especificación del Identificador del caso de prueba. TC-006, Se intenta calcular los atributos para un semestre en el que una de las materias tomadas no tiene un código válido.

Ítems de Prueba. Se prueba el ingreso de la lista de códigos de materias tomadas en un semestre. Si al menos uno de los códigos de las materias tomadas en el semestre es un código no válido no se le asignará valores a los atributos del semestre.

Especificación de entradas:

materias=XXXXZZZZ ICF01099 ICF01107 ICM00604

Especificación de salidas:

total_difficulty=NaN

total_stringency=NaN

total_skewness=NaN

classes_num=NaN

level_mean=NaN

Dependencia de otros casos de prueba. Ninguna

- Caso de prueba TC-007

Especificación del Identificador del caso de prueba. TC-007, Se intenta calcular los atributos para un semestre en el que se toma solo una materia y esta tiene un código válido.

Ítems de Prueba. Se prueba el cálculo de atributos para un semestre en el que se toma solamente una materia. Si solo se ha tomado una materia durante el semestre el cálculo de los atributos se basará en los atributos de dicha materia.

Especificación de entradas:

materias=ICF01099

Especificación de salidas:

total_difficulty=1.347809859

total_stringency=1.874427302

total_skewness=-0.682845274

classes_num=1

level_mean=3.84810126582278

- Caso de prueba TC-008

Especificación del Identificador del caso de prueba. TC-008, Se intenta calcular los atributos para un semestre en el que todas las materias tomadas tienen un código válido.

Ítems de Prueba. Se prueba el cálculo de atributos para un semestre en el que sus materias tienen un código válido.

Especificación de entradas:

materias=ICF01099 ICF01107 ICM00216 ICQ00018

Especificación de salidas:

total_difficulty=4.796833036

total_stringency=4.627092122

total_skewness=-1.373175317

classes_num=4

level_mean=1.025

Dependencia de otros casos de prueba. Ninguna

- Caso de prueba TC-009

  Especificación del Identificador del caso de prueba. TC-009, Se intenta calcular los atributos para un semestre con una lista de códigos de materias vacía.

  Ítems de Prueba. Se probará el ingreso de la lista de códigos de materias vacía. Si no se ingresan códigos de materias no se le asignará valores a los atributos del semestre.

  Especificación de entradas:

  materias=

  Especificación de salidas:

  total_difficulty=NaN

  total_stringency=NaN

  total_skewness=NaN

  classes_num=NaN

  level_mean=NaN

Dependencia de otros casos de prueba. Ninguna

- **Especificación de casos de pruebas para validación de clustering**

  Especificación del diseño de prueba. TDS-03, validación del proceso de clustering para obtener el número indicado de clusters.

  Enfoque específico. Se valida el clustering usando los parámetros del exponente de ponderación, el número de clusters y la matriz de membresía.

  Características a ser probadas. Validar que el exponente de membresía y el número de clusters se encuentre dentro del rango permitido.

  Identificación de pruebas:

  TC-010: Se intenta calcular el índice de validación para un exponente de ponderación menor o igual a uno.

  TC-011: Se intenta calcular el índice de validación para un número de clusters menor a dos.

  TC-012: Se intenta calcular el índice de validación para un exponente de ponderación mayor a uno, y un número de clusters mayor e igual a dos.

- Caso de prueba TC-010

  Especificación del Identificador del caso de prueba. TC-010, Se intenta calcular el índice de validación para un exponente de ponderación menor o igual a uno.

  Ítems de Prueba. Se prueba el ingreso del exponente de ponderación, el cual debe de mantenerse en el rango $(1, \infty)$. Si el parámetro ingresado no se encuentra dentro del rango válido no se le asignará un valor al índice de validación.

  Especificación de entradas:

  m=0

  c=5

  Especificación de salidas:

  v_mpc=NaN

  Dependencia de otros casos de prueba. Ninguna

- Caso de prueba TC-011

  Especificación del Identificador del caso de prueba. TC-011, Se intenta calcular el índice de validación para un número de clusters menor a dos.

  Ítems de Prueba. Se prueba el ingreso del parámetro de número de clusters, el cual debe de mantenerse en el rango $[2, n-1)$ donde n es el número de datos. Si el parámetro

ingresado no se encuentra dentro del rango válido no se le asignará un valor al índice de validación.

Especificación de entradas:

m=1.25

c=1

Especificación de salidas:

v_mpc=NaN

Dependencia de otros casos de prueba. Ninguna

- Caso de prueba TC-012

Especificación del Identificador del caso de prueba. TC-012, Se intenta calcular el índice de validación para un exponente de ponderación mayor a uno, y un número de clusters mayor e igual a dos.

Ítems de Prueba. Se prueba el ingreso del exponente de ponderación y el número de clusters dentro del rango válido.

Especificación de entradas:

m=1.5

c=5

Especificación de salidas:

v_mpc=0.6939626

Dependencia de otros casos de prueba. Ninguna

- **Especificación de casos de pruebas para estimación de riesgo de fallo**

  Especificación del diseño de prueba. TDS-04, estimación de riesgo de fallo para un estudiante en un semestre dado.

  Características a ser probadas. Validar tipos de datos ingresados, datos dentro del rango permitido y campos obligatorios.

  Identificación de pruebas:

  TC-013: Se intenta estimar el riesgo para un estudiante ingresando una matrícula que no es válida, e ingresando en la lista de códigos de materias a tomar en el semestre al menos un código de materia no válido.

  TC-014: Se intenta estimar el riesgo para un estudiante ingresando datos no numéricos como matrícula de un estudiante, además se intenta ingresar un semestre en el que se toma solo una materia y esta no tiene un código válido.

  TC-015: Se intenta estimar el riesgo para un estudiante sin historia académica, y además se ingresa un semestre con códigos de materias válidas.

TC-016: Se intenta estimar el riesgo para un estudiante con historia académica, y además se ingresa un semestre con códigos de materias válidas.

- Caso de prueba TC-013

  Especificación del Identificador del caso de prueba. TC-013, Se intenta estimar el riesgo para un estudiante ingresando una matrícula que no es válida, e ingresando en la lista de códigos de materias a tomar en el semestre al menos un código de materia no válido.

  Ítems de Prueba. Se prueba el ingreso de una matrícula válida, también se prueba el ingreso de la lista de códigos de materias tomadas en un semestre. Si no se ingresa una matrícula válida de un estudiante y al menos uno de los códigos de las materias tomadas en el semestre es un código no válido no se debe permitir la estimación.

  Especificación de entradas:

  matricula=301625248

  materias=XXXXZZZZ ICF01099 ICF01107 ICM00604

  Especificación de salidas:

  riesgo=NaN

  Dependencia de otros casos de prueba. Ninguna

- Caso de prueba TC-014

  Especificación del Identificador del caso de prueba. TC-014, Se intenta estimar el riesgo para un estudiante ingresando datos no numéricos como matrícula de un estudiante, además se intenta ingresar un semestre en el que se toma solo una materia y esta no tiene un código válido.

  Ítems de Prueba. Se probará el ingreso de una matrícula válida, también se prueba el ingreso de la lista de códigos de materias tomadas en un semestre. Si no se ingresa una matrícula válida de un estudiante y al menos uno de los códigos de las materias tomadas en el semestre es un código no válido no se debe permitir la estimación.

  Especificación de entradas:

  matricula=aaaaaaaaa

  materias=XXXXZZZZ ICF01099 ICF01107 ICM00604

  Especificación de salidas:

  riesgo=NaN

  Dependencia de otros casos de prueba. Ninguna

- Caso de prueba TC-015

Especificación del Identificador del caso de prueba. TC-015, Se intenta estimar el riesgo para un estudiante estudiante sin historia académica, y además se ingresa un semestre con códigos de materias válidas.

Ítems de Prueba. Se prueba la estimación de riesgo al tener solo disponible información de una de las dos estructuras, estudiante o semestre. La estimación debe estar basada solamente en la información del semestre.

Especificación de entradas:

matricula=201311332

materias=ICF01099 ICF01107 ICHE00877 ICM00216 ICQ00018

Especificación de salidas:

riesgo=0.597894736842105

Dependencia de otros casos de prueba. Ninguna

- Caso de prueba TC-016

Especificación del Identificador del caso de prueba. TC-016, Se intenta estimar el riesgo para un estudiante con historia académica, y además se ingresa un semestre con códigos de materias válidas.

Ítems de Prueba. Se prueba la estimación de riesgo al tener solo disponible información de una de las dos estructuras, estudiante o semestre.

Especificación de entradas:

matricula=201230771

materias=ICF00703 ICF01149 ICHE00877 ICM00604 ICM01966

Especificación de salidas:

riesgo=0.872093023

Dependencia de otros casos de prueba. Ninguna

5.2. Pruebas de integración

Para efectos de documentación estas pruebas se realizan basadas en el pseudocódigo presentado en la fase de implementación. Las pruebas aquí mencionadas fueron ejecutadas en las últimas fases de desarrollo del prototipo.

5.2.1. Pruebas top-down

Estas pruebas permiten mostrar el nivel de cohesión entre los componentes mediante la observación de la independencia entre componentes proveedores y consumidores; las pruebas *top-down* que fueron realizadas en el prototipo se enfocan de manera general al módulo formado por los tres componentes explicados en

la fase de diseño, para esto cada sub-componente es sometido a pruebas de flujo de datos (*path coverage*) y pruebas combinacionales (*pairwise*), reemplazando las operaciones más básicas, como métodos de clase, con código de prueba para luego extender los resultados a los componentes del módulo.

5.2.2. Pruebas pairwise

Estas pruebas permiten detectar anomalías en los flujos de datos y manejo de excepciones, estas están relacionadas con el código fuente, se llevan a cabo asignando una categoría a cada una de las líneas de código y luego se rastrea el uso de las variables definidas mediante los du-paths (senderos de uso-definición). Dichas pruebas fueron realizadas sobre los componentes de clustering y clasificación, las pruebas serán ilustradas basadas en el pseudocódigo encargado del proceso de clustering, del proceso de clasificación no supervisada y el proceso de extracción de variables descriptivas.

Tabla 23. Pseudocódigo del proceso de clustering en estudiantes

```
1   SET ε to 1·10^-10
2   LIST vData
3   FOR (c,m) IN (2..16]×(1..5]
4           SET p_c, U to CALL fcm(studentsTrainData, c, m, ε)
5           STORE p_c
6           SET V_MPC to 1 - c·(1 - SUM (u_ip)^2 from U / COUNT studentsTrainData) /
```

```
7 |  (c - 1)
8 |           APPEND (V_MPC, m, c) to vData
9 |  END FOR
  |  STORE vData
```

Tabla 24. Categorías de las líneas en el pseudocódigo del proceso de clustering en estudiantes

Categoría			
línea	definición	p-use	c-use
1	ε		
2	vData		
3	c m	c m	
4	p_r U		studentsTrainData c m ε
5			p_r
6	V_{MPC}		c U studentsTrainData
7			V_{MPC} m c
8			
9			vData

Tabla 25. Du-paths en el pseudocódigo del proceso de clustering en estudiantes

du-paths		
start line-end line	c-use	p-use
1→4	ε	
2→9	vData	
3→4	c	c
3→4	m	m

3→6	c	
3→7	m c	
4→5	p_r	
4→6	U	
6→7	V_{MPC}	

Tabla 26. Pseudocódigo de validación del clustering de estudiantes

```
1   SET vMax to 0
2   SET mMax to 0
3   SET cMax to 0
4   FOR each (V_MPC, m, c) IN vData
5           IF vMax ≤ V_MPC
6                   SET vMax to V_MPC
7                   SET mMax to m
8                   SET cMAx to c
9       END IF
10  END FOR
11  STORE (vMax, mMax, cMax)
```

Tabla 27. Categorías de las líneas en el pseudocódigo de validación del clustering de estudiantes

categoría			
línea	definición	p-use	c-use
1	vMax		
2	mMax		
3	cMax		
4	V_{MPC} m c	V_{MPC} m c	vData
5		vMax V_{MPC}	
6			V_{MPC}
7			m
8			c

9			
10			
11		vMax mMax cMax	

Tabla 28. Categorías de las líneas en el pseudocódigo de validación del clustering de estudiantes

du-paths		
start line-end line	c-use	p-use
1→5		vMax
1→11	vMax	
2→11	mMax	
3→11	cMax	
4→4	vData	
4→5		V_{MPC}
4→6	V_{MPC}	
4→7	m	
4→8	c	

Tabla 29. Pseudocódigo del proceso de clustering de semestres

```
1   LIST vData
2   FOR c IN [2..16]
3           SET kmeans to CALL KMeans(c)
4           CALL kmeans.fit(studentsTrainData)
5           SET w_r to kmeans.cluster_centers
6           STORE w_r
7           SET cIDs to kmeans.compute()
8       STORE cIDs
9           SET V_DUNN to CALL dunnIndex(studentsTrainData, kmeans.clusters, c)
10          APPEND (V_DUNN, c) to vData
11  END FOR
12  STORE vData
```

Tabla 30. Categorías de las líneas en el pseudocódigo del proceso de clustering en semestres

categoría			
línea	definición	p-use	c-use
1	vData		
2	c	c	
3	kmeans		c
4			kmean studentsTrainData
5	w_r		
6			w_r
7	cIDs		kmeans
8			cIDs
9	V_{DUNN}		kmeans c
10	vData		V_{DUNN} c
11			
12			

Tabla 31. Du-paths en el pseudocódigo del proceso de clustering de semestres

du-paths		
start line-end line	c-use	p-use
1→10	vData	
2→3		c
2→9	c	
2→10	c	
2→2		c
3→4	kmeans studentsTrianData	
3→5	kmeans	
3→7	kmeans	
3→9	kmeans	
5→6	w_r	
7→8	cIDs	
9→10	V_{DUNN}	

Tabla 32. Pseudocódigo del cálculo del índice de validación de Dunn

```
1   SET maxΔ to MAX pairwiseDistance(studentsTrainData)
2   FOR i IN [1..c]
3            FOR j IN [1..c]
4                        IF j≠i
5                                    SET V_DUNN to
    pairwiseDistance(kmeans.clusters[i], kmeans.clusters[j]) / maxΔ
6                        END IF
7        END FOR
8   END FOR
```

Tabla 33. Categorías de las líneas en el pseudocódigo del cálculo del índice de validación de Dunn

categoría			
línea	definición	p-use	c-use
1	maxΔ		studentsTrainData
2	i	i	c
3	j	j	c
4		i j	
5	V_{DUNN}		kmeans maxΔ i j
6			
7			
8			

Tabla 34. Du-paths en el pseudocódigo del cálculo del índice de validación de Dunn

du-paths		
start line-end line	c-use	p-use
1→5	maxΔ	
2→4	c	i
2→4	c	j
2→5	i	
2→5	j	

Tabla 35. Pseudocódigo de validación del clustering de semestres

```
1   SET vMax to 0
2   SET cMax to 0
3   FOR each (V_DUNN, c) IN vData
4            IF vMax ≤ V_MPC
5                        SET vMax to V_DUNN
6                        SET cMAx to c
7        END IF
8   END FOR
9   STORE (vMax, cMax)
```

Tabla 36. Categorías de las líneas en el pseudocódigo de validación del clustering de semestres

categoría			
línea	definición	p-use	c-use
1	vMax		
2	cMax		
3	V_{DUNN} c	V_{DUNN} c	vData
4		vMax V_{DUNN}	
5	vMax		V_{DUNN}
6	cMax		c
7			
8			
9			vMax cMax

Tabla 37. Du-paths en el pseudocódigo de validación del clustering de semestres

du-paths		
start line-end line	c-use	p-use
1→4		vMax
1→5	V_{DUNN}	
1→9	vMax	
2→6	cMax	
2→9	cMax	

3→4	vData	V_{DUNN}
3→5	V_{DUNN}	V_{DUNN}
3→6	vData	c
4→5		vMax
4→5	vDUNN	vDUNN
3→3	vData	vDUNN
3→3	vData	c

5.2.3. Pruebas path coverage

Estas pruebas permiten identificar si existen *bugs* en el programa de acuerdo al comportamiento del código fuente cuando se ingresan valores a las variables. Para la ilustración de estas pruebas se usó el pseudocódigo detallado en la implementación de los componentes de *clustering* y clasificación. Utilizando dicho pseudocódigo se construyeron los *paths* de pruebas ilustrados por los *dataflow graphs* mostrados en las gráficas.

Tabla 38. Análisis path-coverage en el pseudocódigo del proceso de clustering en estudiantes

```
1   SET ε to 1·10^-10                                              ┐
2   LIST vData                                                     ┘ A
3   FOR (c,m) IN (2..16]×(1..5] ◄── B
4           SET p_c, U to CALL fcm(studentsTrainData, c, m, ε)     ┐
5           STORE p_c                                              │
6           SET V_MPC to 1 - c·(1 - SUM (u_ic)^2 from U / COUNT studentsTrainData) /
7   (c - 1)                                                        │ C
8           APPEND (V_MPC, m, c) to vData                          ┘
9   END FOR ◄── D
    STORE vData ◄── E
```

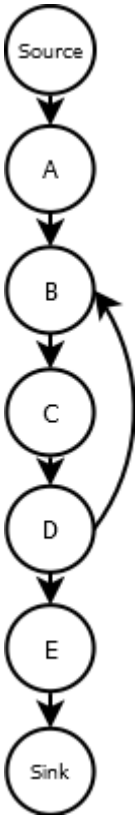


Figura 5.1. Grafo dataflow para el código del proceso de clustering de estudiantes

Tabla 39. Análisis path-coverage en el pseudocódigo de validación del clustering de estudiantes

```
1    SET vMax to 0                          ┐
2    SET mMax to 0                          ├ A
3    SET cMax to 0                          ┘
4            IF vMax < V_MPC ← B
5                                           ┐
6                                           ├ D
7                                           ┘
8        END IF ← E
9    END FOR ← F
10   STORE (vMax, mMax, cMax) ← G
11
```

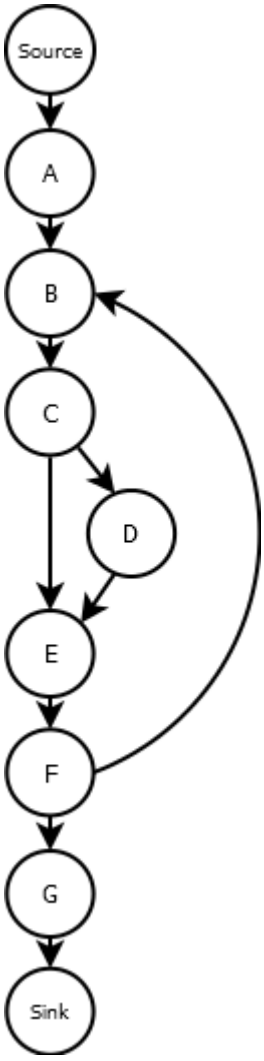


Figura 5.2. Grafo dataflow para el código de validación del clustering de estudiantes

Tabla 40. Análisis path-coverage en el pseudocódigo del proceso de clustering en semestres

1	`LIST vData` ⟵	A
2	`FOR c IN [2..16]` ⟵	B
3	`SET kmeans to CALL KMeans(c)`	C
4	`CALL kmeans.fit(studentsTrainData)`	
5	`SET` w_c `to kmeans.cluster_centers`	
6	`STORE` w_c	
7	`SET cIDs to kmeans.compute()`	

```
8         STORE cIDs
9             SET V_DUNN to CALL dunnIndex(studentsTrainData, kmeans.clusters, c)
10            APPEND (V_DUNN, c) to vData
11    END FOR ← D
12    STORE vData ← E
```

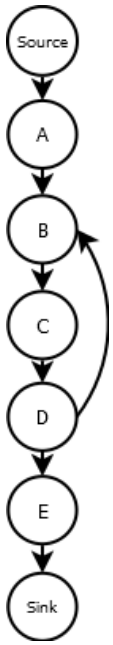


Figura 5.3. Grafo dataflow para el código del proceso de clustering de semestres

Tabla 41. Análisis path-coverage en el pseudocódigo del cálculo del índice de Dunn

```
1   SET maxΔ to MAX pairwiseDistance(studentsTrainData)  ← A
2   FOR i IN [1..c]  ← B
3           FOR j IN [1..c]  ← C
4                   IF j≠i  ← D
5                               SET V_DUNN to                                  ← E
    pairwiseDistance(kmeans.clusters[i], kmeans.clusters[j]) / maxΔ
                    ← F
6                       END IF
7       END FOR  ← G
8   END FOR  ← H
```

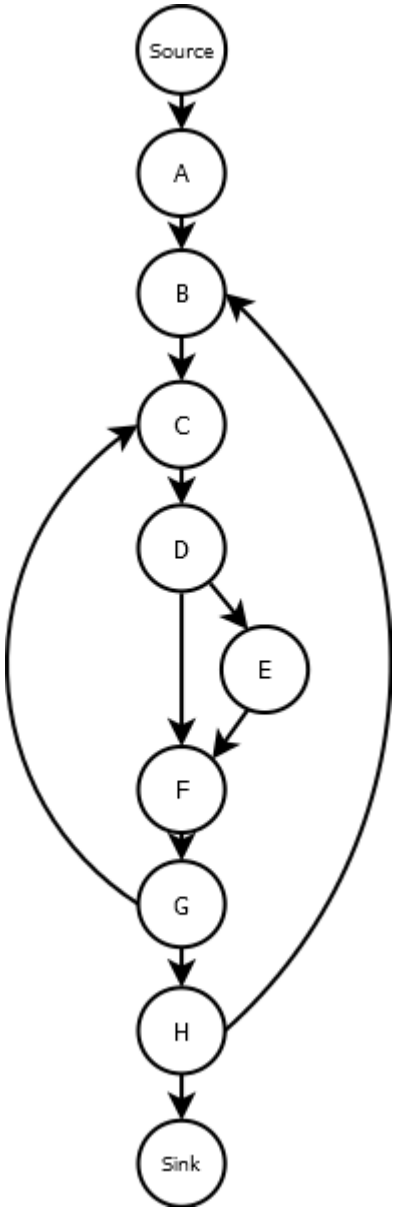


Figura 5.1. Grafo dataflow para el código del cálculo del índice de Dunn

Tabla 42. Análisis path-coverage en el pseudocódigo de validación del clustering en semestres

1	SET vMax to 0	A
2	SET cMax to 0	
3	FOR each (V_{DUNN}, c) IN vData	← B

```
4 |          IF vMax ≤ V_MPC ← C
5 |                          SET vMax to V_DUNN
6 |                          SET cMAx to c   } D
7 |      END IF ← E
8 | END FOR ← F
9 | STORE (vMax, cMax) ← G
```

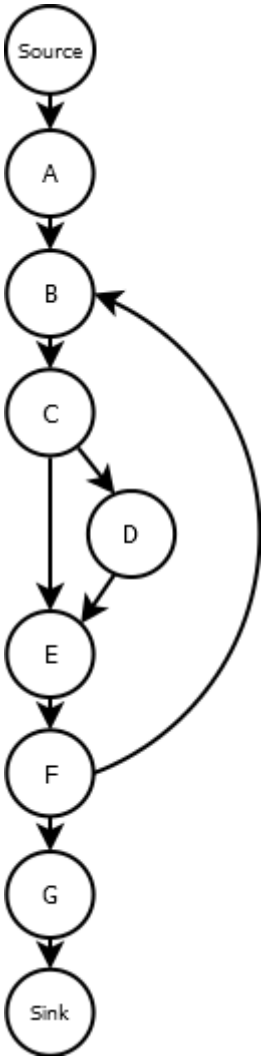


Figura 5.5. Grafo dataflow para el código de validación del clustering de semestres

5.3. Pruebas de rendimiento

El rendimiento fue analizado de forma puntual en líneas de código que requieren un tiempo de procesamiento significativo, dichas pruebas fueron realizadas y registradas en la máquina de desarrollo; los resultados son detallados en la siguiente tabla.

Tabla 43. Cálculo de tiempos medidos en líneas de código específicas de los componentes de clustering y clasificación

Componente	Código	Descripción	Número de lazos	Tiempo por lazo
Componente de clustering	cntr, U, U0, d, Jm, p, fpc = skf.cmeans(data, C, m, error, maxiter)	Invocación a la implementación del algoritmo de FCM	1000	385 ms
Componente de clustering	kmeans.fit(data)	Invocación a la implementación del algoritmo de k-Means	1000	880 ms
Componente de clasificación	U_f, U0_f, d_f, Jm_f, p_f, fpc_f = skf.cmeans_predict(new_student_data.T, p_r, m, error, maxiter)	Cálculo de la función de membresía para nuevos datos de estudiante	1000	642 µs
Componente de clasificación	np.argmax(U_f)	Defusificación	1000	5.13 µs
Componente de clasificación	svc.fit(w_r, categories)	Entrenamiento del SVC para datos de semestres usando los centroides obtenidos en el proceso de k-means	10	1.37 s
Componente de clasificación	cid = svc.predict(new_semester_data)	Cálculo de la categoría usando el clasificador	1000	177 µs

CAPITULO 6

6. DISCUCIÓN DE RESULTADO

6.1. Discusión de experimentos y pruebas

En este capítulo se detallará el resultado del experimento de verificación de estimación de riesgo, este experimento es realizado para probar el nivel de certeza que maneja el modelo implementado, además se realizará una explicación de los resultados y una discusión sobre estos. La verificación del modelo de estimación de riesgo de falla académica se realizó con el score de Brier [55], el cual es una función usada para medir la precisión de un modelo de predicción, a través de un contraste entre la probabilidad predicha y la frecuencia observada; la verificación se realizó sobre el pronóstico de falla en estudiantes que tomaron materias durante los dos términos académicos del 2013, y los casos de reprobación real; para la estimación se utilizaron las

calificaciones de estudiantes inscritos en términos académicos entre los años 1978 y 2012.

El score de Brier puede ser descompuesto dando una perspectiva para analizar el comportamiento del modelo a mayor profundidad [71]; los términos que componen el score de Brier son: incertidumbre, confiabilidad y resolución; la incertidumbre es un término que mide la incertidumbre inherente de que ocurra el evento; mientras que la confiabilidad es un término que mide qué tan cercana está la probabilidad pronosticada a la probabilidad real; finalmente el término de resolución mide qué tanto difiere la probabilidad, dada por los diferentes pronósticos, con el promedio; y usando estos términos se puede calcular el score de *skill* de Brier el cual mide la diferencia entre el score para la predicción y el score de una predicción no cualificada estándar, y es definido como:

$$BSS = \frac{resolution - reliability}{uncertainty} \qquad (6.1)$$

El score de Brier es apropiado para pronóstico binario, es decir, el pronóstico de que ocurra un evento o no, el evento a pronosticar en este caso es la reprobación de al menos una materia durante el semestre; este score mientras menor sea, puede ser interpretado

como una mayor precisión para el modelo [55]; en cuanto a cada término en el que es descompuesto: el término de incertidumbre mientras más cercano a cero sea es interpretado como una mayor certeza de que ocurra el evento; el término de confiabilidad mientras más cercano a cero sea, el pronóstico es más confiable; y el término de resolución mientras mayor sea, es interpretado como una mayor resolución; el score de *skill* mientras más cercano a cero sea, mayor es la habilidad del modelo para pronosticar, pero el *skill* puede alcanzar valores negativos, en cuyo caso se considera al modelo no cualificado [71].

A continuación se hará una descripción de los resultados de la verificación con el score de Brier para el primer y segundo término del 2013, usando el modelo para estimar el riesgo de fallo.

Para el análisis del primer término del año 2013, se encontró un score de Brier de 0.2164, cercano a cero, un score de *skill* positivo de 0.1304; y, con una confiabilidad de 0.0101, cercana a cero; también se encontró una resolución 0.04256 cercana a cero y un porcentaje de incertidumbre de 24.88%; esto quiere decir que para en el primer semestre del 2013 se pudo realizar una estimación en el riesgo de fallo con una certeza del 75.12%.

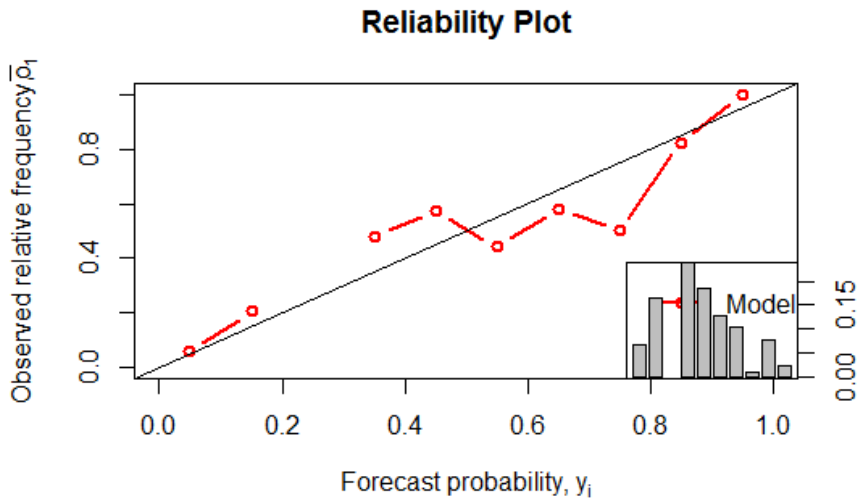


Figura 6.1. Confiabilidad para la estimación de riesgo de falla en estudiantes de Ciencias Computacionales que tomaron materias en el término académico 2013-I

Mientras que en el análisis del segundo término 2013 se encontró un score de Brier de 0.2422, un score de *skill* igual a 0.03113, también positivo, con una confiabilidad de 0.01486 cercana a cero, una resolución de 0.02264 también cercana a cero, y una incertidumbre de 24.99%; esto quiere decir que para el segundo término se encontró que la estimación en el riesgo de fallo posee una certeza del 75.01%.

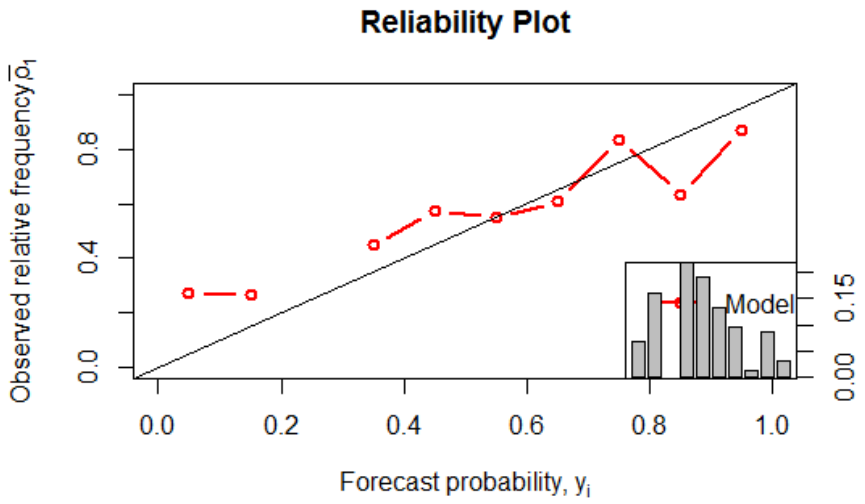


Figura 6.2. Confiabilidad para la estimación de riesgo de falla en estudiantes de Ciencias Computacionales que tomaron materias en el término académico 2013-II

Estos resultados nos dicen que la arquitectura propuesta permite estimar el riesgo de falla para estudiantes con una certeza que bordea el 75% luego del entrenamiento con los datos históricos de los estudiantes, además podemos decir que el modelo posee una confiabilidad significativa en ambos términos; también, se puede apreciar una mayor resolución para la estimación de riesgo de falla en el primer término que la estimación en el segundo término del 2013.

6.2. Limitaciones

La principal limitación con respecto al modelo planteado es que el riesgo de fallo se da por semestre, es decir, que no es una

estimación de falla realizada a las materias específicas, por lo que un estudiante al que se le estime el riesgo de reprobar al menos una materia en el semestre no podrá saber cuál es la materia que posee el mayor riesgo potencial, aunque se puede mitigar esta limitante mostrando las medidas de dificultad o rigurosidad calculadas por materia.

CONCLUSIONES Y RECOMENDACIONES

En este trabajo se ha propuesto un modelo de manejo de incertidumbre que permita estimar el riesgo de fallo en materias durante el semestre, este modelo está basado en prototipos y clasificación no supervisada, soportada por algoritmos de clustering, estos algoritmos de clustering permiten construir prototipos dentro de un conjunto de datos, a través de valores de tipicidad asociados a cada elemento; dichos prototipos representan los elementos más distintivos de cada cluster; además, pueden ser usados como datos de entrenamiento para la clasificación no supervisada de nuevos datos. A continuación se enumerarán las respectivas conclusiones a las que se ha llegado.

1. El proceso de clustering debe de ser validado en los casos en los que no se conozca de manera previa el número de categorías que posea el conjunto de datos, aunque en el clustering difuso existen varias de estas medidas de validación, no existe una regla estricta que exprese los casos

o circunstancias en las que se deba usar una medida de validación u otra, a lo que se concluye que una forma de encontrar el índice de validación que mejor se ajuste al data set, es el diseño de un experimento en el que se pueda verificar visualmente el proceso de clustering junto con los parámetros estimados por el proceso de validación.

2. La selección de las características que serán usadas en el proceso de clustering juegan un papel muy importante, ya que deben de aportar a la semántica que proveen los prototipos obtenidos en este proceso, en este caso las medidas en el contexto académico a seleccionar representaban la carga académica semestral y rendimiento de un estudiante, así los prototipos representaban elementos distintivos en el rendimiento de estudiantes y la carga académica semestral, y también representaban una categoría de estudiantes con rendimiento similar y semestres con carga académica similar.
3. El rendimiento de un estudiante pudo ser descrito gracias a las medidas que representan las habilidades ganadas al aprobar materias, estas medidas permitieron definir la similitud entre estudiantes en el proceso de clustering difuso; la carga académica semestral pudo ser descrita mediante las medidas de dificultad, rigurosidad y distribución de notas que están asociadas a una materia, y así determinar las categorías de semestres a las que un estudiante se puede afrontar.

4. Con el diseño propuesto para el prototipo de estimación, se obtuvo un porcentaje de certeza significativo en resultados experimentales sugiriendo que el modelo planteado, para la arquitectura y configuración, posee un nivel de respuesta aceptable en la estimación del riesgo de reprobación en al menos una materia durante el semestre para estudiantes de las carreras de Ingeniería en Ciencias Computacionales de la ESPOL; esto quiere decir que si dicho módulo es implementado e implantando en el sistema de Consejerías Académicas podría proveer de información para retroalimentar a los profesores consejeros con respecto al riesgo de falla durante el periodo de pre-registro.

A continuación se enumerarán las recomendaciones y sugerencias para trabajos futuros.

1. Para trabajos futuros se sugiere explorar la generalización de este modelo de estimación basado en prototipos difusos, ya que el caso de estudio en este trabajo está orientado al contexto académico, pero sería de gran interés aplicarlo a casos que no necesariamente se encuentren relacionados al ámbito académico.
2. Además los resultados obtenidos utilizando el modelo propuesto muestran un porcentaje de certeza significativo, a pesar de que los datos provistos para entrenamiento solo poseían información académica, por lo que una estrategia a futuro es la incorporación de

otros factores relacionados con carga extra-universitaria, datos socioeconómicos o información demográfica, para así mejorar el nivel de precisión de la estimación del riesgo de reprobación.

ANEXOS

ANEXO I - RESULTADOS EXPERIMENTALES DEL CONTRASTE DE ÍNDICES DE VALIDACIÓN

C	M	V_{pbmf}	V_{pe}	V_{mpc}
5	1.25	0.29647987	0.20296823	0.86077604
8	1.25	0.02965929	0.24372415	0.85675315
9	1.25	0.04842931	0.25551907	0.85385448
4	1.25	0.36523795	0.19496194	0.85272689
7	1.25	0.14917038	0.23988202	0.85163307
6	1.25	0.19640131	0.23095378	0.85047072
3	1.25	1.68335261	0.17409231	0.84808947
10	1.25	0.02412507	0.27206835	0.84545792
11	1.25	0.02342518	0.28046259	0.84428366
12	1.25	0.00595307	0.29278058	0.83817088
13	1.25	0.02967291	0.31715077	0.82618085
15	1.25	0.01000711	0.32472122	0.82525132
14	1.25	0.00841342	0.32305633	0.82465077
2	1.5	6.44991237	0.20016591	0.76064533
5	1.5	0.33628395	0.48314184	0.6939626
6	1.5	0.07605365	0.54421224	0.67947109
4	1.5	0.21392314	0.44681035	0.67586878
3	1.5	1.88102543	0.38900781	0.66848607
7	1.5	0.14707346	0.59524791	0.66843691
8	1.5	0.09343564	0.65577766	0.65167771
9	1.5	0.0700622	0.72086171	0.62910059
2	1.75	4.56936331	0.30763999	0.62740432
10	1.5	0.01785303	0.75304897	0.62298104
11	1.5	0.02982488	0.76722623	0.62135605
12	1.5	0.01286603	0.8136857	0.60697829
13	1.5	0.00688487	0.8374273	0.6002287
14	1.5	0.0025909	0.86792901	0.59350133
15	1.5	0.00714398	0.89221652	0.58728429
4	1.75	0.62028271	0.66716553	0.52559487
5	1.75	0.38431388	0.76434907	0.52347953
6	1.75	0.09115104	0.85725676	0.5089692
3	1.75	1.75293917	0.57311855	0.50886804
2	2	5.9121752	0.39813775	0.50325614

7	1.75	0.1334255	0.97955377	0.46933939
8	1.75	0.05053424	1.0773022	0.4407783
9	1.75	0.04862472	1.15830178	0.42056567
10	1.75	0.03431453	1.21542092	0.41161721
12	1.75	0.00872459	1.29245938	0.40422877
4	2	0.88360744	0.84576196	0.39945532
2	2.25	5.61330323	0.46741372	0.39940041
11	1.75	0.02602895	1.2750594	0.39889627
13	1.75	0.01136604	1.35168742	0.39274024
14	1.75	0.00923007	1.39949786	0.38530805
3	2	1.86409643	0.70998214	0.38466194
5	2	0.40669643	0.9905275	0.38087681
15	1.75	0.0039295	1.4439178	0.37951748
6	2	0.21173927	1.11546608	0.36429636
7	2	0.0881309	1.25358763	0.33019174
2	2.5	4.88784256	0.51868222	0.31722235
8	2	0.07384542	1.37537085	0.29940082
4	2.25	0.89257445	0.9822765	0.29935439
3	2.25	1.79230378	0.80824822	0.29179688
9	2	0.04549423	1.46130506	0.28485089
5	2.25	0.39266398	1.15156197	0.27844728
10	2	0.01814796	1.55096826	0.26748769
11	2	0.01016265	1.60730813	0.26315981
2	2.75	5.07473184	0.55640467	0.2537148
6	2.25	0.2043815	1.31345243	0.25111485
12	2	0.01691708	1.68766342	0.2500956
13	2	0.01299994	1.74537738	0.24509474
14	2	0.00532769	1.81420448	0.23475178
15	2	0.00734204	1.87401542	0.22468141
7	2.25	0.11381252	1.46137975	0.22410986
3	2.5	1.70939955	0.87831348	0.22352515
4	2.5	0.80611228	1.08395131	0.22314199
5	2.5	0.32399574	1.26598224	0.2057228
2	3	4.84629012	0.5843447	0.20494606
8	2.25	0.05627808	1.57957767	0.20409823
9	2.25	0.04423456	1.6733912	0.19428931
10	2.25	0.02534928	1.77298548	0.18020584
11	2.25	0.01550775	1.83074834	0.17907007
6	2.5	0.18863438	1.4434474	0.1776191

3	2.75	1.62658636	0.92857046	0.17349865
2	3.25	4.64331833	0.60528819	0.16739273
4	2.75	0.65889035	1.16001163	0.16567475
12	2.25	0.01511936	1.94266692	0.16035385
13	2.25	0.00796812	1.99333209	0.16033006
14	2.25	0.0078725	2.06006988	0.15448292
5	2.75	0.35151529	1.34781873	0.15408546
7	2.5	0.10454662	1.60222712	0.15315821
15	2.25	0.00669727	2.1146586	0.15002435
2	3.5	4.71435747	0.62120787	0.13826121
8	2.5	0.06490025	1.727474	0.13793798
3	3	1.54897168	0.96506413	0.13665529
6	2.75	0.18133698	1.52756097	0.13160803
9	2.5	0.03906263	1.83848914	0.1267175
4	3	0.6737888	1.21963471	0.12079646
5	3	0.33273142	1.40691886	0.11726636
10	2.5	0.02865493	1.94048834	0.116288
2	3.75	4.30149679	0.63348373	0.11544454
11	2.5	0.01884619	2.01553461	0.11512089
7	2.75	0.09994797	1.68704326	0.11179466
12	2.5	0.01475806	2.09167266	0.11041476
3	3.25	1.54661975	0.99197065	0.10924653
13	2.5	0.00764255	2.17387272	0.10350645
8	2.75	0.06187564	1.81339859	0.10095427
14	2.5	0.00798306	2.23511269	0.10067951
6	3	0.17344116	1.58749296	0.09945572
2	4	4.54969512	0.64308295	0.09738511
15	2.5	0.00522757	2.31415945	0.09478119
9	2.75	0.03815981	1.92560305	0.09281088
5	3.25	0.31559495	1.45021102	0.09071561
4	3.25	0.63272053	1.26182709	0.08944728
3	3.5	1.41501565	1.01213568	0.088594
10	2.75	0.02734566	2.03041687	0.08470941
7	3	0.09846845	1.74661661	0.08357874
2	4.25	4.46465491	0.65068914	0.08293774
12	2.75	0.01393736	2.18580192	0.0813376
11	2.75	0.0188428	2.12596947	0.07812098
13	2.75	0.0102314	2.26237084	0.07677428
6	3.25	0.16558674	1.63085569	0.07670529

8	3	0.06070114	1.87395526	0.07580044
14	2.75	0.00766135	2.33577441	0.07295847
3	3.75	1.52648145	1.02749716	0.07281381
2	4.5	3.90874893	0.65679106	0.0712588
5	3.5	0.33083096	1.48248286	0.0712527
15	2.75	0.00585696	2.40130013	0.0705609
9	3	0.0395742	1.98601003	0.07024253
4	3.5	0.60182705	1.28940788	0.0691087
10	3	0.026625	2.09045801	0.06426969
7	3.25	0.10382912	1.78915917	0.06410126
2	4.75	3.80163369	0.66174259	0.06172305
3	4	1.51134568	1.03938405	0.0605855
6	3.5	0.15923584	1.66277346	0.06034343
11	3	0.01850518	2.18567632	0.05956423
8	3.25	0.05885839	1.91722225	0.05840911
5	3.75	0.32646364	1.50701833	0.05670135
12	3	0.01324551	2.27268293	0.05556417
9	3.25	0.03958671	2.02905689	0.05474083
4	3.75	0.55998703	1.30905732	0.05472573
15	3	0.00559893	2.47076115	0.05321166
13	3	0.00971151	2.3494058	0.05245874
3	4.25	1.33321434	1.04871805	0.05097957
14	3	0.00729212	2.41962352	0.05053988
7	3.5	0.09494542	1.82002476	0.05045832
10	3.25	0.02584272	2.13313863	0.05035325
6	3.75	0.15356125	1.68672629	0.04831789
11	3.25	0.01795344	2.2278378	0.04691978
13	3.25	0.00940544	2.37724297	0.0462808
5	4	0.31968294	1.52606525	0.04558772
14	3.25	0.00728071	2.44696195	0.04486011
4	4	0.60903912	1.3233913	0.04431607
8	3.5	0.05508945	1.95364642	0.04392706
12	3.25	0.01341509	2.31484961	0.04385604
9	3.5	0.03943672	2.06058147	0.0437402
3	4.5	1.22158334	1.05614725	0.04333592
15	3.25	0.00539245	2.51556385	0.04305137
10	3.5	0.025134	2.16434277	0.04063161
7	3.75	0.08753343	1.84314039	0.04044643
6	4	0.14732674	1.7050463	0.03930344

11	3.5	0.0217037	2.25878825	0.03796956
3	4.75	1.45639969	1.06213411	0.0371804
5	4.25	0.2894917	1.54114302	0.03692859
4	4.25	0.52632713	1.33406259	0.03662234
15	3.5	0.00535711	2.54695306	0.03641531
12	3.5	0.01318611	2.34474797	0.03599232
13	3.5	0.00907221	2.42099864	0.03488675
8	3.75	0.05631325	1.97789489	0.03486017
10	3.75	0.02428401	2.1873841	0.03347887
14	3.5	0.00685916	2.49652224	0.0332955
7	4	0.11437715	1.8608266	0.03295643
14	3.75	0.00662761	2.50398607	0.03256444
6	4.25	0.14398009	1.71929826	0.03241881
11	3.75	0.02314239	2.28057732	0.03207141
12	3.75	0.01308777	2.36520583	0.03118735
9	3.75	0.03298135	2.09569706	0.03079951
4	4.5	0.57949677	1.34221974	0.03077664
13	3.75	0.0122226	2.44308195	0.03039323
5	4.5	0.29846641	1.55320837	0.03010603
8	4	0.05131127	1.9949197	0.02848974
7	4.25	0.0824369	1.87459437	0.02722559
6	4.5	0.15586396	1.7305557	0.02707201
4	4.75	0.44781945	1.34862854	0.02620715
9	4	0.03161445	2.11306784	0.0250822
5	4.75	0.24142064	1.563111	0.0245888
8	4.25	0.05007024	2.00993602	0.02317817
6	4.75	0.13516368	1.73969622	0.02278422
7	4.5	0.10200018	1.88548457	0.02277266
10	4	0.0229977	2.21950772	0.02209763
9	4.25	0.03044686	2.12680364	0.02064193
11	4	0.01452657	2.31649272	0.01962925
8	4.5	0.05333396	2.01972399	0.0195639
7	4.75	0.07836477	1.89421403	0.01926124
10	4.25	0.02859401	2.23320566	0.01819092
12	4	0.0103612	2.40483367	0.01765646
9	4.5	0.03168613	2.13757558	0.01725417
15	3.75	0.00473864	2.61266152	0.01691108
8	4.75	0.04711455	2.02891586	0.01638898
11	4.25	0.01403876	2.3297203	0.01618888

13	4	0.00756176	2.48618856	0.01598015
10	4.5	0.02887652	2.24407505	0.01518497
14	4	0.00620857	2.55962861	0.01487385
9	4.75	0.03235808	2.14617086	0.01458115
12	4.25	0.01423301	2.4179736	0.01452188
15	4	0.00457838	2.62932243	0.01372276
11	4.5	0.01754051	2.34016684	0.01351416
13	4.25	0.00800217	2.49911243	0.01311855
10	4.75	0.01870246	2.25228066	0.01284474
14	4.25	0.00540432	2.57285586	0.01215047
12	4.5	0.00966678	2.42825486	0.01209259
11	4.75	0.01370868	2.34852347	0.01141123
15	4.25	0.00411415	2.6423887	0.01120345
13	4.5	0.00916948	2.50894474	0.01094325
12	4.75	0.01463139	2.43654826	0.01018857
14	4.5	0.00598333	2.58440464	0.00990811
15	4.5	0.0039413	2.65284163	0.00926187
13	4.75	0.00687015	2.51733549	0.00920575
14	4.75	0.00516975	2.59126594	0.00850358
15	4.75	0.00520619	2.66086916	0.00781022

ANEXO II – GRÁFICO DE RADAR PARA LAS MEDIDAS DE HABILIDADES EN LOS CLUSTERS DE ESTUDIANTES

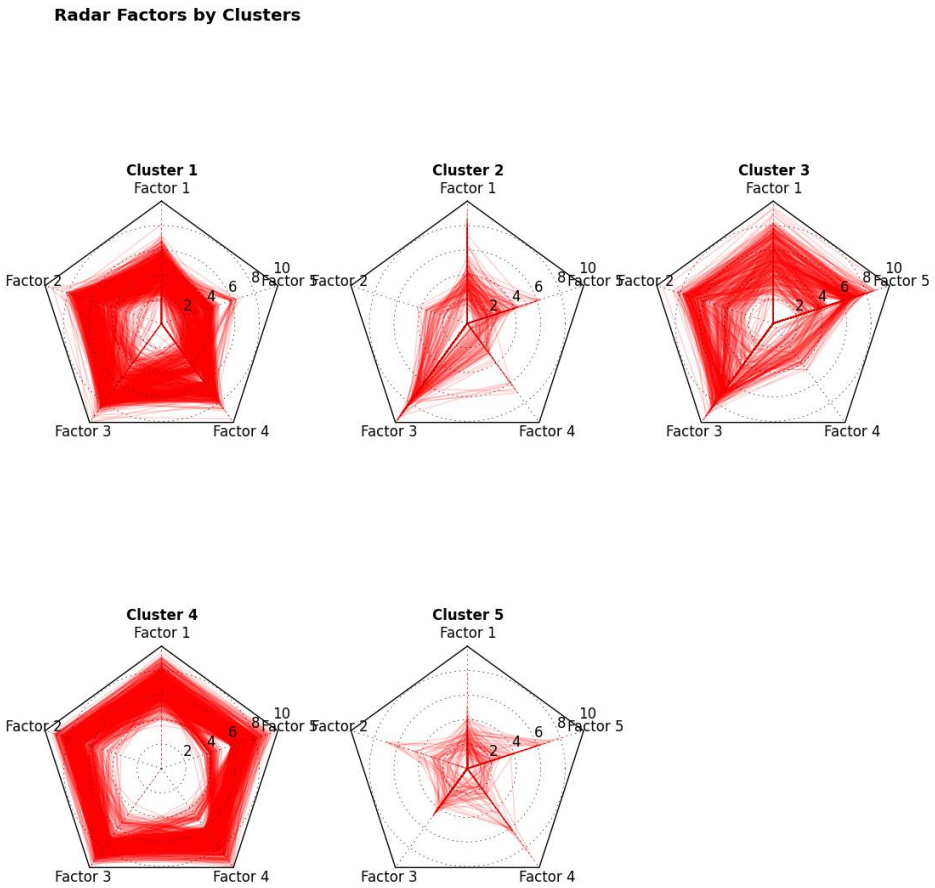


Figura Anexo II.1. Gráfico de radar para las medidas de habilidades en los clusters de estudiantes resultado de FCM usando un m=1.25 y un c=5

ANEXO III – RESULTADOS EXPERIMENTALES DE LA VERIFICACIÓN DE LA PREDICCIÓN DE RIESGO DE FALLO PARA EL PRIMER TÉRMINO DEL 2013

The forecasts are probabilistic, the observations are binary.
Sample baseline calculated from observations.
Brier Score (BS) = 0.2164
Brier Score - Baseline = 0.2488
Skill Score = 0.1304
Reliability = 0.0101
Resolution = 0.04256
Uncertainty = 0.2488

ANEXO IV – RESULTADOS EXPERIMENTALES DE LA VERIFICACIÓN DE LA PREDICCIÓN DE RIESGO DE FALLO PARA EL SEGUNDO TÉRMINO DEL 2013

The forecasts are probabilistic, the observations are binary.
Sample baseline calculated from observations.
Brier Score (BS) = 0.2422
Brier Score - Baseline = 0.2499
Skill Score = 0.03113
Reliability = 0.01486
Resolution = 0.02264
Uncertainty = 0.2499

BIBLIOGRAFÍA

[1] Kaufman, L., & Rousseeuw, P. J. *Finding groups in data: an introduction to cluster analysis* (Vol. 344). John Wiley & Sons, 2009.

[2] Bezdek, J. C. (1981). *Pattern recognition with fuzzy objective function algorithms*. Kluwer Academic Publishers, 1981.

[3] Apte, C., Grossman, E., Pednault, E. P., Rosen, B. K., Tipu, F. A., & White, B. Probabilistic estimation-based data mining for discovering insurance risks. *IEEE Intelligent Systems*, *14*(6), 1999, p. 49-58.

[4] Grossman, E., Pednault, E., Rosen, B., Tipu, F., & White, B. *Insurance risk modeling using data mining technology*. IBM Thomas J. Watson Research Division, 1998.

[5] Milton, O. GPA Tyranny. In *National Forum: Phi Kappa Phi Journal.* Vol. 68, No. 3, 1988, p. 43-45.

[6] Caulkins, J. P., Larkey, P. D., & Wei, J. Adjusting GPA to reflect course difficulty, 1996.

[7] Méndez, G., Ochoa, X., & Chiluiza, K. Techniques for data-driven curriculum analysis. In *Proceedins of the Fourth International Conference on Learning Analytics And Knowledge*. ACM, 2014, p. 148-157.

[8] Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J., & Tibshirani, R. *The elements of statistical learning* Vol. 2, No. 1. New York: springer, 2009, p. 148-157.

[9] MacQueen, J. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium*

on mathematical statistics and probability Vol. 1, No. 14, 1967, p. 281-297.

[10] Arthur, D., & Vassilvitskii, S. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007, p. 1027-1035.

[11] Frey, B. J., & Dueck, D. Clustering by passing messages between data points. *science*, *315*(5814), 2007, p. 972-976.

[12] Givoni, I. E., & Frey, B. J. A binary variable model for affinity propagation. *Neural computation*, *21*(6), 2009, p. 1589-1600.

[13] Macnaughton-Smith, P., Williams, W. T., Dale, M. B., & Mockett, L. G. Dissimilarity analysis: a new technique of hierarchical sub-division, 1964.

[14] Bezdek, J. C., Ehrlich, R., & Full, W. FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, *10*(2), 1984, p. 191-203.

[15] Krishnapuram, R., & Keller, J. M. A possibilistic approach to clustering. *Fuzzy Systems, IEEE Transactions on*, *1*(2), 1993, p. 98-110.

[16] Lesot, M. J. Similarity, typicality and fuzzy prototypes for numerical data. In *6th European Congress on Systems Science, Workshop Similarity and resemblance* Vol. 94, 2005, p. 95-96.

[17] Lesot, M. J., Rifqi, M., & Bouchon-Meunier, B. Fuzzy prototypes: From a cognitive view to a machine learning principle. In*Fuzzy Sets and Their Extensions: Representation, Aggregation and Models*. Springer Berlin Heidelberg, 2008, p. 431-452.

[18] Rosch, E., & Mervis, C. B. Family resemblances: Studies in the internal structure of categories. *Cognitive psychology*, *7*(4), 1975, p. 573-605.

[19] Rifqi, M., Berger, V., & Bouchon-Meunier, B. Discrimination power of measures of comparison. *Fuzzy sets and systems*, *110*(2), 2000, p. 189-196.

[20] Bouchon-Meunier, B., Rifqi, M., & Bothorel, S. Towards general measures of comparison of objects. *Fuzzy sets and systems*, *84*(2), 1996, p. 143-153.

[21] Rifqi, M. Constructing prototypes from large databases. In*International conference on Information Processing and Management of Uncertainty in knowledge-based systems, IPMU'96*, 1996.

[22] Weber-Lee, R., Barcia, R. M., Martins, A., & Pacheco, R. C. Using typicality theory to select the best match. In *Advances in Case-Based Reasoning* (pp. 445-459). Springer Berlin Heidelberg, 1996.

[23] Lesot, M. J., Mouillet, L., & Bouchon-Meunier, B. Fuzzy prototypes based on typicality degrees. In *Computational Intelligence, Theory and Applications* (pp. 125-138). Springer Berlin Heidelberg, 2005.

[24] Doring, C., Borgelt, C., & Kruse, R. Fuzzy clustering of quantitative and qualitative data. In *Fuzzy Information, 2004. Processing NAFIPS'04. IEEE Annual Meeting of the* Vol. 1, IEEE, 2004, p. 84-89.

[25] Pal, N. R., Pal, K., & Bezdek, J. C. A mixed c-means clustering model. In *Fuzzy Systems, 1997, Proceedings of the Sixth IEEE International Conference on* Vol. 1, IEEE, 1997, p. 11-21.

[26] Bezdek, J. C. *Pattern recognition with fuzzy objective function algorithms*. Kluwer Academic Publishers, 1981.

[27] Windham, M. P. Cluster validity for fuzzy clustering algorithms.*Fuzzy Sets and Systems*, *5*(2), 1981, p. 177-185.

[28] Dave, R. N. Validating fuzzy partitions obtained through c-shells clustering. *Pattern Recognition Letters*, *17*(6), 1996, p. 613-623.

[29] Kim, Y. I., Kim, D. W., Lee, D., & Lee, K. H. A cluster validation index for GK cluster analysis based on relative degree of sharing. *Information Sciences*, *168*(1), 2004, p. 225-242.

[30] Chen, M. Y., & Linkens, D. A. Rule-base self-generation and simplification for data-driven fuzzy models. In *Fuzzy Systems, 2001. The 10th IEEE International Conference on* Vol. 1, IEEE, 2001, p. 424-427.

[31] Fukuyama, Y., & Sugeno, M. A new method of choosing the number of clusters for the fuzzy c-means method. In *Proc. 5th Fuzzy Syst. Symp* Vol. 247, 1989, p. 247-250.

[32] Xie, X. L., & Beni, G. A validity measure for fuzzy clustering. *IEEE Transactions on pattern analysis and machine intelligence*, *13*(8), 1991, p. 841-847.

[33] Kwon, S. H. Cluster validity index for fuzzy clustering. *Electronics Letters*, *34*(22), 1998, p. 2176-2177.

[34] Tang, Y., Sun, F., & Sun, Z. Improved validation index for fuzzy clustering. In *American Control Conference, 2005. Proceedings of the 2005* IEEE, 2005, p. 1120-1125.

[35] Zahid, N., Limouri, M., & Essaid, A. A new cluster-validity for fuzzy clustering. *Pattern recognition*, *32*(7), 1999, p. 1089-1097.

[36] Gath, I., & Geva, A. B. Unsupervised optimal fuzzy clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *11*(7), 1989, p. 773-780.

[37] Wu, K. L., & Yang, M. S. A cluster validity index for fuzzy clustering. *Pattern Recognition Letters*, *26*(9), 2005, p. 1275-1291.

[38] Tsekouras, G. E., & Sarimveis, H. A new approach for measuring the validity of the fuzzy c-means algorithm. *Advances in Engineering Software*, *35*(8), 2004, p. 567-575.

[39] Rezaee, M. R., Lelieveldt, B. P., & Reiber, J. H. A new cluster validity index for the fuzzy c-mean. *Pattern recognition letters*, *19*(3), 1998, p. 237-246.

[40] Sun, H., Wang, S., & Jiang, Q. FCM-based model selection algorithms for determining the number of clusters. *Pattern recognition*, *37*(10), 2004, p. 2027-2037.

[41] Kim, D. W., Lee, K. H., & Lee, D. On cluster validity index for estimation of the optimal number of fuzzy clusters. *Pattern Recognition*, *37*(10), 2004, p. 2009-2025.

[42] Pakhira, M. K., Bandyopadhyay, S., & Maulik, U. Validity index for crisp and fuzzy clusters. *Pattern recognition*, *37*(3), 2004, p. 487-501.

[43] Bouguessa, M., & Wang, S. R. A new efficient validity index for fuzzy clustering. In *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on* Vol. 3, IEEE, 2004, p. 1914-1919.

[44] Hassar, H., & Bensaid, A. Validation of fuzzy and crisp c-partitions. In *Fuzzy Information Processing Society, 1999. NAFIPS. 18th International Conference of the North American,* IEEE, 1999, p. 342-346.

[45] Xie, Y., Raghavan, V. V., Dhatric, P., & Zhao, X. A new fuzzy clustering algorithm for optimally finding granular prototypes. *International Journal of Approximate Reasoning*, *40*(1), 2005, p. 109-124.

[46] Yu, J., & Li, C. X. Novel cluster validity index for FCM algorithm. *Journal of Computer Science and Technology*, *21*(1), 2006, p. 137-140.

[47] Pal, N. R., & Bezdek, J. C. On cluster validity for the fuzzy c-means model. *Fuzzy Systems, IEEE Transactions on*, *3*(3), 1995, p. 370-379.

[48] Rhee, H. S., & Oh, K. W. (1996, September). A validity measure for fuzzy clustering and its use in selecting optimal number of clusters. In *Fuzzy Systems, 1996., Proceedings of the Fifth IEEE International Conference on* Vol. 2, IEEE, 1996, p. 1020-1025.

[49] Barash, Y., & Friedman, N. Context-specific Bayesian clustering for gene expression data. *Journal of Computational Biology*, *9*(2), 2002, p. 169-191.

[50] Cho, S. B., & Yoo, S. H. Fuzzy Bayesian validation for cluster analysis of yeast cell-cycle data. *Pattern Recognition*, *39*(12), 2006, p. 2405-2414.

[51] Rhee, H. S., & Oh, K. W. A validity measure for fuzzy clustering and its use in selecting optimal number of clusters. In *Fuzzy Systems, 1996, Proceedings of the Fifth IEEE International Conference on* Vol. 2, IEEE, 1996, p. 1020-1025.

[52] Chong, A., Gedeon, T. D., & Koczy, L. T. A hybrid approach for solving the cluster validity problem. In *Digital Signal Processing, 2002. DSP 2002. 2002 14th International Conference on* Vol. 2, IEEE, 2002, p. 1207-1210.

[53] Wang, W., & Zhang, Y. On fuzzy cluster validity indices. *Fuzzy sets and systems*, *158*(19), 2007, p. 2095-2117.

[54] IEEE Standard for Software and System Test Documentation - Redline," *IEEE Std 829-2008 (Revision of IEEE Std 829-1998) – Redline*, 2008.

[55] Brier, G. W. Verification of forecasts expressed in terms of probability.*Monthly weather review*, *78*(1), 1950, p. 1-3.

[56] Van Der Walt, S., Colbert, S. C., & Varoquaux, G. The NumPy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, *13*(2), 2011, p. 22-30.

[57] McKinney, W. *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. “O’Reilly Media, Inc.”, 2012.

[58] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., & Duchesnay, É. Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, *12*, 2011, p. 2825-2830.

[59] Warner, J., Scikit-Fuzzy, https://github.com/scikit-fuzzy/scikit-fuzzy/blob/master/README.md, fecha de consulta enero de 2015

[60] McGaha, V., & Fitzpatrick, J. Personal and social contributors to dropout risk for undergraduate students. *College Student Journal*, *39*(2), 2005, p. 287-288.

[61] Dunn, J. C. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, 1973.

[62] ESPOL, Webservices de información académica, https://ws.espol.edu.ec/saac/wsandroid.asmx, fecha de consulta febrero de 2015

[63] Bhardwaj, B. K., & Pal, S. Data Mining: A prediction for performance improvement using classification. *arXiv preprint arXiv:1201.3418* ,2012.

[64] Naser, S. A., Zaqout, I., Ghosh, M. A., Atallah, R., & Alajrami, E. Predicting Student Performance Using Artificial Neural Network: in the Faculty of Engineering and Information Technology, 2015.

[65] Pandey, M., & Taruna, S. A Multi-Level Classification Model Pertaining to the Student's Academic Performance Prediction. *International Journal of Advances in Engineering & Technology*, *7*(4), 2014.

[66] ESPOL, Estadísticas de Crecimiento, http://www.espol.edu.ec/espol/main.jsp?urlpage=registrados.jsp, fecha de consulta agosto de 2014

[67] ESPOL, Directorio de Ex-Alumnos, http://www.espol.edu.ec/alumni/A/pagina/1/directorio.aspx, fecha de consulta agosto de 2014

[68] Bradburn, E. M. Short-Term Enrollment in Postsecondary Education: Student Background and Institutional Differences in Reasons for Early Departure, 1996-98. Postsecondary Education Descriptive Analysis Reports, 2002.

[69] Bates, L., Luster, T., & Vandenbelt, M. Factors related to social competence in elementary school among children of adolescent mothers. *Social Development*, *12*(1), 2003, p. 107-124.

[70] ESPOL, Consejerías Estudiantiles, https://www.fiec.espol.edu.ec/index.php/Comunidad/com-consejerias-estudiantiles-18-04-2012.html, fecha de consulta septiembre de 2014

[71] Murphy, A. H. A new vector partition of the probability score. *Journal of Applied Meteorology*, *12*(4), 1973, p. 595-600.