Pontificia Universidad Católica de Valparaíso

Facultad de Ingeniería

Escuela de Ingeniería Informática

AUTOMATIZACIÓN DE LA CLASIFICACIÓN DE DIAGNÓSTICOS MÉDICOS.

VÍCTOR HUGO MATHEU RODRIGUES

Profesor Guía: **Rodrigo Alfaro Arancibia**

Carrera: **Ingeniería Civil en Informática**

Abril 2017

Agradecimientos

Agradezco a Dios que me dio la fuerza, la fe y destinó mi vida para llegar a la universidad en donde he logrado mi desarrollo personal.

Agradezco a mi padre que me enseño, me acompaño en mis primeros pasos y ahora está en los cielos,

Agradezco a mi madre y mis hermanos que siempre me entendieron, siendo siempre tan diferente.

Y agradezco a mis profesores por su sacrificio y por su dedicación personal y su entrega total.

Resumen

La información y el conocimiento, es la base de investigación y desarrollo en las nuevas tecnologías TICs (Tecnologías de la información y comunicación). Cada día, las tecnologías son más importantes en la vida cotidiana y más áreas se suman aplicándolas. En el área clínica Las técnicas de análisis, extracción, clasificación y ranking de documento clínicos, juegan un papel transcendental pero oculto para la mayoría de seres humanos. Después de los sistemas bancarios, los sistemas de salud son la segunda gran red privada de información que se transmite. Por lo tanto, hay un gran reto en el área de la minería de datos para el trabajo de datos clínicos. El presente documento abarca técnicas y enfoques para poder clasificar las historias clínicas dentro de un conjunto de diagnósticos CIE-10. De esta manera, poder construir un clasificador autónomo que de soporte al personal médico en la toma de decisiones.

Palabras Claves: Clasificación de Texto, Clasificador Cadena, Clasificación Múltiple, Clasificación Binaria, Maquinas de Soporte Vectorial, Naive Bayes, Aprendizaje de Maquinas.

Abstract

Information and knowledge, is the basis of research and development in the new technologies ICT (Information and communication technologies). Every day, technologies are more important in everyday life and more areas are added by applying them. In the clinical area The techniques of analysis, extraction, classification and classification of clinical documents play a transcendental but hidden role for most human beings. After banking systems, health systems are the second major private information network to be transmitted. Therefore is a great challenge in the area of data mining for the work of clinical data. This document covers techniques and approaches for classifying clinical records within a set of ICD-10 diagnoses. In this way, we can build an autonomous classifier that supports medical personnel in decision making.

Keywords: Classification of Text, Classifier Chain, Multiple Classification, Binary Classification, Support Vector Machines, Naive Bayes, Machine Learning.

Índice

Lista de Figuras

Lista de Tablas

1. Introducción

Desde que surgieron los primeros documentos en formato digital, la clasificación automática de textos adquirió mucha relevancia para muchos investigadores. Se trata de asignar automáticamente documentos a categorías predefinas.

En la actualidad, se está utilizando en muchas aplicaciones tales como clasificación de páginas web, filtrado de spam, etc. Sin embargo, por parte del área de la salud hay muchos proyectos donde puede ser utilizada las técnicas de clasificación.

En el área de la salud, existen múltiples procesos en los cuales debe participar un paciente con un profesional médico. Desde un procedimiento manual asociado directamente con el paciente, hasta el registro o traspaso de un formulario a un sistema de información.

Dentro de los procedimientos médicos, uno de los principales es el registro de la observación médica del profesional. En los sistemas de información hospitalaria (HIS – Hospital Information System) es muy concurrente que se haga obligatorio para cerrar el proceso de la atención médica el ingreso de la observación y la asignación del diagnóstico.

Para el diagnóstico, se utiliza una clasificación denominada CIE-10, acrónimo de la Clasificación Internacional de Enfermedades*,* ésta es la décima versión correspondiente a la versión en español de la ICD(International Statistical Classification of Diseases and Related Health Problems).

La clasificación y codificación CIE-10, determina las enfermedades y una amplia variedad de signos, síntomas, hallazgos anormales, denuncias, circunstancias sociales y causas externas de daños y/o enfermedad.

Por la gran cantidad de códigos CIE-10 existentes para el cierre de atención, el profesional debe conocer previamente el código, el no ser así, genera una gran tarea para encontrar el correspondiente. Este problema ocurre en todos los HIS, en donde es necesario ingresar el código CIE-10 para el cierre de atención con el paciente.

Por este problema, las entidades de salud no registran el código CIE-10. Esto es un grave problema en el historial clínico, ya que los gobiernos y organizaciones privadas buscan la estandarización de la ficha electrónica del paciente.

En este trabajo, se realiza una investigación sobre la relación que existe entre la observación médica, la historia del paciente y el diagnóstico CIE-10 y aplicar técnicas para la clasificación de texto.

1.1 Objetivos

En este trabajo, se busca a través de técnicas de clasificadores de texto encontrar la relación desde la observación médica, a un conjunto de diagnósticos CIE -10 que den soporte al profesional médico con la mejor precisión posible.

1.1.1 Objetivo General

Determinar cuáles son las diferencias entre el método Clasificación Cadena (CC) Binaria, SMO de WEKA y Naive Bayes de WEKA, para las historias médicas propuestas, sobre las representaciones de TF-IDF y TF-RFL.

1.1.2 Objetivos Específicos

- Investigar y analizar las técnicas de clasificación de texto Etiquetas Múltiples y la Clasificación de Cadena Binaria sobre la representación TF-IDF y TF-RFL.
- Investigar y analizar la técnica de SMO de WEKA sobre la representación TF-IDF y TF-RFL.
- Investigar y analizar la técnica de Naive Bayes de WEKA sobre la representación TF-IDF y TF-RFL.
- Comparar y determinar las diferencias entre los métodos propuestos y determinar cuál de ellos entrega el mejor resultado.

2. Marco Teórico

En los últimos diez años, la automatización basada en el contenido de los documentos ha ganado un lugar destacado en los sistemas de información, en gran parte debido a la mayor disponibilidad de los documentos en formato digital y la consiguiente necesidad por parte de los usuarios para acceder a ellas en formas flexible. La Categorización de texto (conocido como Text Classification –TC) tiene una larga historia, que data desde los años 60, pero no fue hasta los años 90 que se convirtió en una materia importante de investigación, y esto sucede gran parte por la disponibilidad de hardware más potente. Hoy en día las TICs se utilizan en muchos aplicativos, que van desde la indexación automática de documentos basados en un vocabulario controlado, hasta la generación de metadatos automatizados [7].

Hasta finales de los 80, el método más popular de la clasificación de texto, consistía en definir manualmente un conjunto de operadores expertos de codificación, para clasificar los documentos en virtud de las categorías presentadas. Sin embargo, desde a principios de los años 90 este enfoque perdió relevancia a favor de la Máquina de Aprendizaje (Machine Learning - ML). En este último, existe un proceso inductivo que construye automáticamente un clasificador de texto de aprendizaje, desde un conjunto de documentos previamente clasificados. Las ventajas de este enfoque, es que se tiene una mejor precisión comparado con expertos humanos y existe un ahorro considerable de términos y procesos que se debían realizarse en la antigua metodología [7].

Actualmente, la clasificación de texto es una disciplina en la encrucijada con las máquinas de aprendizaje y la recuperación de información (Retrivial Informaticon- IR) y todas como tal comparten una serie de características con otras tareas tales como la información del conocimiento, la extracción de texto y la minería de texto (Text Mining). Todavía hay un debate en donde ese encuentra la frontera de estas disciplinas y sus metodologías. Tentativamente podemos observar que la minería de texto se utiliza para las tareas que mediante el análisis de grandes cantidades de texto se encuentren patrones que se utilizan para extraer la información útil y según este enfoque la Clasificación de Texto es una instancia de Minería de Texto [7].

3. Clasificación Internacional de Enfermedades Décima Versión (CIE - 10) y Tecnologías de Clasificación

En esta sección, se expondrán los códigos CIE-10 y la utilización de clasificadores en el área médica [8].

La CIE fue publicada por la Organización Mundial de la Salud. Se utiliza a nivel internacional para fines estadísticos relacionados con morbilidad y mortalidad, los sistemas de reintegro y soportes de decisión automática en medicina. Este sistema está diseñado para promover la comparación internacional de la recolección, procesamiento, clasificación y presentación de estas estadísticas. La CIE es la clasificación central de la WHO Family of International Classifications (WHO-FIC) (en español, la Familia de Clasificaciones Internacionales de la OMS) [8].

La lista CIE-10 tiene su origen en la «Lista de causas de muerte», cuya primera edición editó el Instituto Internacional de Estadística en 1893. La OMS se hizo cargo de la misma en 1948, en la sexta edición, la primera en incluir también causas de morbilidad. A la fecha, la lista en vigor es la décima, y la OMS sigue trabajando en ella [8].

La CIE-10 se desarrolló en 1992 y su propósito, fue rastrear estadísticas de mortalidad. La OMS publica actualizaciones menores anuales y actualizaciones mayores cada tres años [8].

En EE.UU se añadió el anexo con el sistema de clasificación de procedimientos o ICD-10-PCS [8].

Cada condición de salud puede ser asignada a una categoría y recibir un código de hasta seis caracteres de longitud (en formato de X00.00). Cada una de tales categorías puede incluir un grupo de enfermedades similares [8].

Tabla 3.1 : Secciones de la Codificación CIE-10.

Capítulo	Códigos	Título
I	A00-B99	Ciertas enfermedades infecciosas y parasitarias
II	C00-D48	Neoplasias
III	D50-D89	Enfermedades de la sangre y de los órganos hematopoyéticos y otros trastornos que afectan el mecanismo de la inmunidad
IV	E00-E90	Enfermedades endocrinas, nutricionales y metabólicas
V	F00-F99	Trastornos mentales y del comportamiento
VI	G00-G99	Enfermedades del sistema nervioso
VII	H00-H59	Enfermedades del ojo y sus anexos
VIII	H60-H95	Enfermedades del oído y de la apófisis mastoides
IX	I00-I99	Enfermedades del sistema circulatorio
X	J00-J99	Enfermedades del sistema respiratorio
XI	K00-K93	Enfermedades del aparato digestivo
XII	L00-L99	Enfermedades de la piel y el tejido subcutáneo
XIII	M00-M99	Enfermedades del sistema osteomuscular y del tejido conectivo
XIV	N00-N99	Enfermedades del aparato genitourinario
XV	O00-O99	Embarazo, parto y puerperio
XVI	P00-P96	Ciertas afecciones originadas en el periodo perinatal
XVII	Q00-Q99	Malformaciones congénitas, deformidades y anomalías cromosómicas
XVIII	R00-R99	Síntomas, signos y hallazgos anormales clínicos y de laboratorio, no clasificados en otra parte
XIX	S00-T98	Traumatismos, envenenamientos y algunas otras consecuencias de causa externa
XX	V01-Y98	Causas extremas de morbilidad y de mortalidad

4. Metodología

El reconocimiento de patrones es la ciencia que se ocupa de los procesos sobre ingeniería, computación y matemáticas relacionados con objetos físicos o abstractos, con el propósito de extraer información que permita establecer propiedades entre conjuntos de dichos objetos [9].

Los patrones se obtienen a partir de los procesos de segmentación, extracción de características y descripción, dónde cada objeto queda representado por una colección de descriptores. El sistema de reconocimiento, debe asignar a cada objeto su categoría o clase (conjunto de entidades que comparten alguna característica que las diferencias del resto). Para poder reconocer los patrones se siguen los siguientes procesos:

- Adquisición de datos
- Extracción de características
- Toma de decisiones

El punto esencial del reconocimiento de patrones es la clasificación: se quiere clasificar una señal dependiendo de sus características. Señales, características y clases pueden ser de cualquiera forma, por ejemplo se puede clasificar imágenes digitales de letras en las clases «A» a «Z» dependiendo de sus píxeles o se puede clasificar ruidos de cantos de los pájaros en clases de órdenes aviares dependiendo de las frecuencias[9].

4.1 Clasificación Automática de Texto

Durante los últimos veinte años, la rápida expansión que ha experimentado Internet en todo el mundo, ha hecho posible que el acceso a todo tipo de información sea una tarea de baja complejidad. Cada vez es mayor el número de fuentes de contenidos y el volumen de datos que se tiene al alcance, y este crecimiento explosivo de documentos disponibles complica su exploración y análisis. Por consiguiente, son necesarios nuevos métodos que ayuden a los usuarios a filtrar y estructurar la información relevante. Por ello, poder organizar la información de forma automática ha pasado a ser una tarea de vital importancia y llevar a cabo una gestión eficiente de la información se ha convertido en algo imprescindible. Por este motivo cada vez son más necesarias herramientas que puedan automatizar esta clasificación [12].

La clasificación automática de texto consiste en un conjunto de algoritmos, técnicas y sistemas capaces de asignar un documento a una o varias categorías o grupos de documentos, construidos según su afinidad temática. Para ello, se emplean técnicas de Aprendizaje Automático (ML, Machine Learning) y de Procesamiento de Lenguaje Natural (NLP, Natural Language Processing) [12].

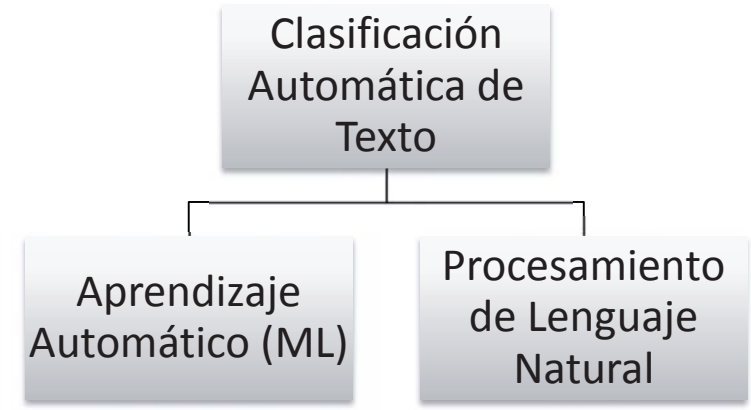

Figura 4.1 : Clasificación Automática de Texto

El Procesamiento de Lenguaje Natural (PLN) estudia los problemas inherentes al procesamiento y manipulación de lenguajes naturales, haciendo uso de computadoras. Pretende adquirir conocimiento sobre el modo en que los humanos entienden y utilizan el lenguaje, de tal forma que se pueda llevar a cabo el desarrollo de herramientas y técnicas para conseguir que los ordenadores puedan entenderlo y manipularlo. Sus fundamentos residen en un conjunto muy amplio de disciplinas: ciencias de la información, computadores, lingüística, matemáticas, ingeniería eléctrica y electrónica, inteligencia artificial y robótica, psicología, etc. Existe un gran número de aplicaciones donde el PLN resulta de gran utilidad (traducción máquina, procesamiento y resumen de textos escritos en lenguaje natural, interfaces de usuario, reconocimiento de voz, etc.) [12].

Para el diseño de la función de clasificación, se pueden emplear diferentes técnicas de aprendizaje, debiendo disponer para ello de un conjunto de documentos (conjunto de entrenamiento), que previamente han sido clasificados dentro de una determinada categoría. Estos algoritmos de aprendizaje o entrenamiento, requieren una representación estructurada de los documentos. La más empleada es la basada en el modelo de espacio vectorial, donde cada documento se transforma en un vector de palabras clave, a las que se les asigna un peso en función de la importancia o relevancia que éstas representen dentro del documento. Una vez que el clasificador ha sido entrenado con el correspondiente grupo de textos, su efectividad se evalúa comparando las categorías que ha asignado a los documentos del set de prueba con las que éstos ya tenían asignadas. Este esquema permite alcanzar una precisión comparable a la obtenida por expertos humanos, reduciendo así los costes de mano de obra [12].

Algunos ejemplos de los entornos en los que se emplea la clasificación automática son: indexación automática de textos, filtrado de textos, clasificación de páginas Web, filtrado de correos electrónicos (*spam*), o clasificación de noticias [12].

4.2 Tipos de Clasificadores

En el esquema siguiente presentan las distintas características que puede presentar un clasificador, según diferentes puntos de vista:

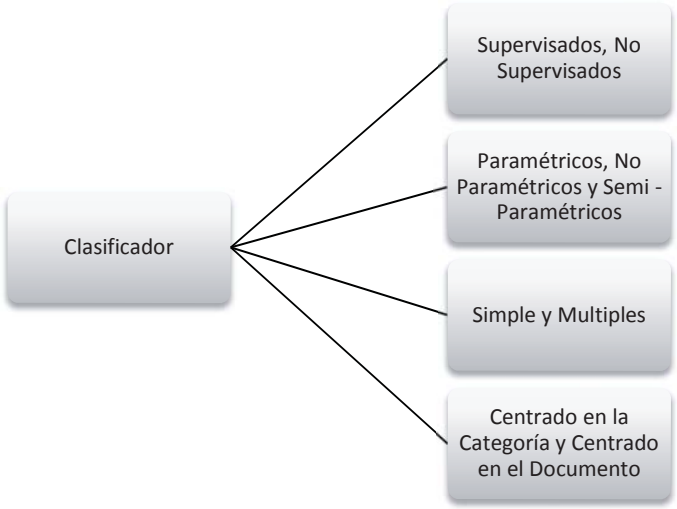

Figura 4.2 : Tipos de Clasificadores

4.2.1 Clasificación Supervisada

La Clasificación supervisada, parte de una serie de categorías conceptuales prediseñadas a priori, se encarga de asignar cada documento a la categoría correspondiente. Requiere la elaboración manual o intelectual del conjunto de categorías. Además, es necesaria una fase de entrenamiento por parte del clasificador [12].

El objetivo que se persigue en los clasificadores supervisados es el siguiente: elaborar un patrón representativo para cada una de las categorías entrenadas y aplicar alguna función que permita estimar la similitud entre el documento a clasificar y cada uno de estos patrones. Aquel patrón o patrones que presenten más concordancias con el documento indicarán la categoría o categorías a las que pertenece el mismo. El proceso de elaboración de los patrones necesita un conjunto de documentos previamente clasificados y se conoce como aprendizaje o entrenamiento [12].

Clasificación no supervisada: No existen categorías previas o cuadros de clasificación establecidos a priori. Los documentos se clasifican en función de su contenido, de forma automática, sin asistencia manual. Es una segmentación o agrupamiento automático, en inglés conocido como clustering[12].

4.2.2 Clasificación Caracterizada por Parámetros

Clasificación paramétrica: El entrenamiento de un clasificador se emplea el *set* de entrenamiento para estimar o aprender los parámetros del modelo. El *set* de test que contiene documentos a clasificar se emplea para determinar la capacidad de generalización del clasificador [12].

Clasificación no paramétrica: Se subdivide en dos categorías. La primera está basada en patrones y se obtiene una descripción de cada categoría en términos de un patrón, normalmente en forma de vector de términos con peso, como por ejemplo el clasificador Rocchio. La clasificación de los documentos se realiza en función de las similitudes existentes entre cada documento y los distintos patrones [Bacan, Pandzic, &Gulija, 2005]. La segunda categoría está basada en ejemplos y los documentos se clasifican según las similitudes que presenten con ejemplos del conjunto de entrenamiento. El clasificador más conocido es el del vecino más cercano (*KNN, K*-NearestNeighbour) [12].

4.2.3 Clasificación Simple y Clasificación Múltiple

Clasificación simple: cada documento tiene una única categoría. Se trata de una clasificación donde las categorías no se solapan. Un caso especial es la clasificación binaria, donde cada documento pertenece a una categoría o a su complementaria.

Clasificación múltiple: cada documento puede recibir un número variable de categorías. En este caso, las categorías sí se pueden solapar [10].

4.2.4 Clasificación Centrada en la Categoría y en el Documento

Una vez construido el clasificador existen dos formas en las que puede ser utilizado, teniendo en cuenta el hecho de que el conjunto de categorías C o el conjunto de documentos D puede que no se encuentren disponibles de forma completa desde el comienzo [10].

Clasificación Centrada en la Categoría (CPC, Category-Pivoted Classification) dado el documento se pueden encontrar todas las categorías dentro de las cuales se puede clasificar [10].

Clasificación Centrada en el Documento (DPC, Document-Pivoted Classification) dada la categoría debemos encontrar todos los documentos que pueden ser clasificados dentro de ella. Adecuada cuando una nueva categoría se añade al conjunto después de que varios documentos ya hayan sido clasificados y es necesario que dichos documentos se vuelvan a tener en cuenta para una posible clasificación [10].

4.3 Técnicas de Clasificación Automática de Textos

4.3.1 Algoritmos Probabilísticos

Se basan en la teoría probabilística, en especial en el teorema de Bayes, el cual permite estimar la probabilidad de un suceso a partir de la probabilidad de que ocurra otro suceso, del cual depende el primero. El algoritmo más conocido, y también el más simple, es el denominado Naive Bayes, que estima la probabilidad de que un documento pertenezca a una categoría. Dicha pertenencia depende de la posesión de una serie de características, de cada una de las cuales se conoce la probabilidad de que aparezcan en los documentos que pertenecen a la categoría en cuestión. Naturalmente, dichas características son los términos que conforman los documentos, y tanto su probabilidad de aparición en general, como la probabilidad de que aparezcan en los documentos de una determinada categoría, pueden obtenerse a partir de los documentos de entrenamiento; para ello se utilizan las frecuencias de aparición en la colección de entrenamiento [13].

Cuando las colecciones de aprendizaje son pequeñas, pueden producirse errores al estimar dichas probabilidades. Por ejemplo, cuando un determinado término no aparece nunca en esa colección de aprendizaje, pero aparece en los documentos a categorizar. Esto implica la necesidad de aplicar técnicas de suavizado, a fin de evitar distorsiones en la obtención de las probabilidades [13].

Con dichas probabilidades, obtenidas de la colección de entrenamiento, podemos estimar la probabilidad de que un nuevo documento, dado que contiene un conjunto determinado de términos, pertenezca a cada una de las categorías. La más probable, obviamente, es a la que será asignado [13].

4.3.2 Algoritmos de Rocchio

El llamado algoritmo de Rocchio, se aplica en la realimentación de consultas. Una vez formulada y ejecutada una primera consulta, el usuario examina los documentos que el clasificador ha devuelto, y determina cuáles le resultan relevantes y cuáles no. Con estos datos, el sistema genera automáticamente una nueva consulta, basándose en los documentos que el usuario señaló como relevantes o no relevantes. En este contexto, el algoritmo de Rocchio, proporciona un sistema para construir el vector de la nueva consulta, re-calculando los pesos de los términos de ésta y aplicando un coeficiente a los pesos de la consulta inicial, otro a los de los documentos relevantes y otro distinto a los de los no relevantes [14].

En el ámbito de la categorización, el mismo algoritmo de Rocchio proporciona un sistema para construir los patrones de cada una de las clases o categorías de documentos. Así, partiendo de una colección de entrenamiento, previamente categorizada de forma manual, y aplicando el modelo vectorial, se pueden construir vectores patrón para cada una de las

categorías, considerando como ejemplos positivos los documentos de entrenamiento de esa categoría, y como ejemplos negativos los de las demás categorías [14].

Una vez que se tienen los patrones de cada una de las clases, el proceso de entrenamiento o aprendizaje está completado. Para categorizar nuevos documentos, simplemente se estima la similitud entre el nuevo documento y cada uno de los patrones. El que presenta un índice mayor indica la categoría a la que se debe asignar ese documento [14].

4.3.3 Algoritmos del Vecino más Próximo y Variantes

El algoritmo del vecino más próximo (*Nearest Neighbour, NN*), es uno de los más sencillos de implementar. La idea básica es como sigue: si se calcula la similitud entre el documento a clasificar y cada uno de los documentos de entrenamiento, aquél de éstos más parecido estará indicando a qué clase o categoría se debe asignar el documento que se desea clasificar [15].

Una de las variantes más conocidas de este algoritmo es la del *k-nearestneighbour*o (*kNN),* que consiste en tomar los *k* documentos más parecidos, en lugar de sólo el primero. Como en esos *k* documentos los habrá, presumiblemente, de varias categorías, se suman los coeficientes de los de cada una de ellas. La que más puntos acumule, será la candidata idónea. El *kNN* une a su sencillez una eficacia notable. Obsérvese que el proceso de entrenamiento no es más que la indexación o descripción automática de los documentos, y que tanto dicho entrenamiento como la propia categorización pueden llevarse a cabo con instrumentos bien conocidos y disponibles para cualquiera. De otra parte, numerosas pruebas experimentales han mostrado su eficacia. *KNN* parece especialmente eficaz cuando el número de categorías posibles es alto, y cuando los documentos son heterogéneos y difusos [15].

Es un método de clasificación no paramétrico, ya que no se hace ninguna suposición distribucional acerca de las variables predictivas. Para inferir la categoría de un ejemplo desconocido, el algoritmo compara ese ejemplo con todos los ejemplos de entrenamiento, calculando la distancia entre ellos. A continuación, la clase mayoritaria de entre los *k* ejemplos más similares al de entrada es la categoría inferida para el mismo. Generalmente se usa la distancia Euclidiana:

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{|C|}(x_{ik} - x_{jk})^2} \quad (4.3.3)$$

4.3.4 Árboles de Clasificación

Es uno de los métodos de aprendizaje inductivo supervisado no paramétrico más utilizado. Como forma de representación del conocimiento, los árboles de clasificación destacan por su sencillez. A pesar de que carecen de la expresividad de las redes semánticas o de la lógica de primer orden, su dominio de aplicación no está restringido a un ámbito concreto sino que pueden utilizarse en diversas áreas: diagnóstico médico, juegos, predicción meteorológica, control de calidad, etc. [16].

Un árbol de clasificación, es una forma de representar el conocimiento obtenido en el proceso de aprendizaje inductivo. Puede verse como la estructura resultante de la partición recursiva del espacio de representación a partir del conjunto de prototipos (documentos). Esta partición recursiva se traduce en una organización jerárquica del espacio de representación que puede modelarse mediante una estructura de tipo árbol. Cada nodo interior contiene una pregunta sobre un atributo concreto (con un hijo por cada posible respuesta) y cada nodo hoja se refiere a una decisión (clasificación) [16].

La clasificación de patrones se realiza en base a una serie de preguntas sobre los valores de sus atributos, comenzando por el nodo raíz y siguiendo el camino determinado por las respuestas a las preguntas de los nodos internos, hasta llegar a un nodo hoja. La etiqueta asignada a esta hoja es la que se asignará al patrón a clasificar [16].

Aprendizaje: Consiste en la construcción del árbol a partir de un conjunto de prototipos. Constituye la fase más compleja, y la que determina el resultado final. A esta fase se dedica la mayor parte de la atención [16].

Clasificación: Consiste en el etiquetado de un patrón, X, independiente del conjunto de aprendizaje. Se trata de responder a las preguntas asociadas a los nodos interiores, utilizando los valores de los atributos del patrón X. Este proceso se repite desde el nodo raíz hasta alcanzar una hoja, siguiendo el camino impuesto por el resultado de cada evaluación [16].

5. Algoritmo de Aprendizaje Computacional

5.1 Clasificación en Cadena (Classifier Chain CC)

En esta sección se explicará el modelo de cadena. Se comenzará con la base del modelo, el método binario y luego se explicará la Clasificación Cadena (CC).

5.1.1 Método Binario de Clasificación (BM)

A pesar de las desventajas difundidas en el entorno del método binario, sus ventajas raramente son conocidas. El método binario es sencillo e intuitivo, la suposición de independencia etiquetada hace que sea adecuado a los contextos en que nuevos ejemplos puede no necesariamente ser relevantes para las etiquetas conocidas o donde las relaciones pueden cambiar con el sello de los datos de prueba, incluso las etiquetas de un conjunto se pueden alterar de forma dinámica, esto es ideal para aprendizajes activos y escenarios de flojo de datos [19].

Sin embargo, la ventaja más importante y relevante ampliamente en el método binario, es su baja complejidad computacional en comparación con otros métodos. Dado un número constante de ejemplos, el método binario tiene un tamaño que funciona linealmente cuando el tamaño del conjunto de las etiquetas L es conocido. Este conjunto está acotado por $|L| < |X|$ donde X es el espacio de características. Si L es muy grande, o no se define antes de la clasificación, el problema es mejor enfocarlo con otras técnicas que dividan el problema [19].

Otro método fundamental de transformación es el método de combinación de etiqueta o método de alimentación de ajuste CM. La base de este método es la combinación de todas las etiquetas para formar un solo problema de etiquetas atómicas. El conjunto formado representará a todos los sub-conjuntos de etiquetas originales. Cada (x, S) se transforma en (x, I) donde I es la etiqueta atómica que representa un subconjunto de etiquetas distintas. De esta manera, los métodos CM toman directamente en cuenta las correlaciones de etiquetas. Una desventaja de estos métodos, sin embargo, es el tiempo [19].

Aunque BM implica $|L|$ problemas de etiqueta única, cada problema solo afecta a dos clases. Dependiendo del conjunto de datos, CM podría lidiar con miles o decenas de combinaciones de clases. Otros métodos de transformación resultan en un solo problema, una producir decisiones que impliquen al menos $|L|$ clases y estos puede llevar a una mayor complejidad lineal [19].

Entonces BM separa |L| problemas binarios, bajo ciertas condiciones, si es concebible ejecutar cada problema por separado, ya sea en paralelo o en serie, por lo que solo requieren |D| instancias en memoria más |L| iteraciones o procesadores [19].

5.1.2 Modelo de Clasificador de Cadena (Classifier Chain CC)

El modelo de clasificador cadena (CC) implica |L| clasificadores binarios como en el BM. Los clasificadores están vinculados a lo largo de una cadena en la que cada uno de los clasificadores dan la relevancia a un problema binario asociado con la etiqueta $l_j \epsilon L$. La característica del espacio de cada uno de los miembros en la cadena se extiende con las asociaciones 0/1 de etiquetas con todos los miembros previos en la cadena. El entrenamiento se describe en la figura 2. Recordando la notación por ejemplo (x, S), donde S es un subconjunto de L y se encuentra representado por un vector binario $(l_1, l_{2,\ldots} l_{|L|}) \in \{0,1\}^{|L|}$ y x es una instancia del vector[19].

```
TRAINING(D = {(x_1, S_1), ··· , (x_n, S_n)})
1  for j ∈ 1 ··· |L|
2       do ▷ single-label transformation and training
3          D' ← {}
4          for (x, S) ∈ D
5               do D' ← D' ∪ ((x, l_1, ··· , l_{j−1}), l_j)
6          ▷ train C_j to predict binary relevance of l_j
7          C_j : D' → l_j ∈ {0, 1}
```

Figura 5.1: Algoritmo CC en fase de entrenamiento.

En consecuencia, una cadena $C_1,\ldots..,C_{|L|}$ binaria de clasificadores es formada por cada clasificador C_J ,que es responsable de aprender y predecir la asociación binaria de la etiqueta L_j dada por la característica del espacio, aumentada por todas las predicciones de relevancia binaria que se encuentran antes de la cadena de$l_{1,\ldots} l_{|j-1|}$. El proceso de clasificación comienza C_1 y se propaga a lo largo de la cadena: C1 determinada por $Pr(l_1|x)$ y todos los siguientes clasificadores $C_2,\ldots..,C_{|L|}$los predice $Pr(l_j|x_i, l_{1,\ldots} l_{j-1})$[19].

```
CLASSIFY(x)
1  Y ← {}
2  for j ← 1 to |L|
3       do Y ← Y ∪ (l_j ← C_j : (x, l_1, ··· , l_{j−1}))
4  return (x, Y) ▷ the classified example
```

Figura 5.2: Algoritmo CC en la fase de predicción para probar la instancia ***x***

El método de encadenamiento pasa la información de las etiquetas entre los clasificadores, lo que permite tener las correlaciones de las etiquetas y por lo tanto superar así el problema que sufre BM y además reteniendo las ventajas del método BM

Sin embargo, en términos de complejidad computacional CC puede estar muy cerca de BM, en función del número total de etiquetas. BM tiene como complejidad O($|L| \times f(|X|,|D|)$), donde $f(|X|,|D|)$ es la complejidad del aprendizaje. Usando la misma notación, la complejidad de CC O($|L| \times f(|X| + |L|,|D|)$) [19].

5.1.3 Conjunto Clasificador de Cadena (Ensembles of Classifier Chains – ECC)

Las clasificadoras cadenas son bien conocidas por su efecto de aumentar la precisión generalmente y superación de paralelismo. Esto ha tenido buenos resultados en problemas de múltiple etiqueta [19].

Los métodos binarios de vez en cuando se refieren a los métodos de conjuntos porque implican múltiples modelos binarios. Ninguno de estos modelos es múltiple etiqueta [19].

ECC encadena mCC clasificadores, $C_{1,}C_{2,}...,C_{m}$. Cada C_k es encadenado con:

- Una cadena ordenada (de L elementos) aleatoriamente.
- Aleatoriamente un sub-conjunto de K.

Por lo tanto, un modelo C_k es probable que sea único y capaz de dar ML diferentes de predicciones. Estas predicciones se suman por la etiqueta y cada etiqueta contenga un número de votos definidos. Luego existe un umbral o rango que se utiliza para seleccionar las etiquetas más populares que forman el conjunto final [19].

Cada k_{th} modelo individual predictor del vector $y_k=(l_1,\ldots,l_{|L|}) \in \{0,1\}^{|L|}$. Las cantidades que almacena el vector $W=(\lambda_1,\ldots,\lambda_{|L|}) \in R^{|L|}$, tal que cada $\lambda_j = \sum_{k=1}^{m} lj \epsilon y_k$. Cada $\lambda_j \epsilon W$ representa la cantidad de votaciones para una etiqueta $l_j \epsilon L$[19].

5.2 Naive Bayes

El clasificador Naive Bayes (Bayes Ingenuo), se construye usando el conjunto de entrenamiento para estimar la probabilidad de cada clase dados los valores de atributos (palabras) del documento de una nueva instancia. Usando el teorema de Bayes para estimar las probabilidades:

$$p(c_j|d) = \frac{P(c_j)P(d|c_j)}{P(d)} \quad (5.2.1)$$

El denominador en la ecuación anterior no distingue entre categorías y puede ser eliminado. Este método, asume que los atributos son condicionalmente independientes, dada la clase. Esto simplifica los cálculos.

$$p(c_j|d) = \mathrm{P}(c_j)\prod_{i=1}^{M}\mathrm{P}(d_i|c_j) \quad (5.2.3)$$

Una estimación $K(c_j)$ para $P(c_j)$ puede ser calculada de la fracción de documentos de entrenamiento que es asignada a la clase c_j

$$K(c_j) = \frac{N_j}{N} \quad (5.2.4)$$

Donde N_j es el número de documentos de entrenamiento para los cuales la clase es c_j y N es el número total de documentos de entrenamiento.

Una estimación $K(d_i|c_j)$ para $P(d_i|c_j)$ *está dada por:*

$$K(di|cj\ ) = \frac{1 + N_{ij}}{M + \sum_{k=1}^{M} N_{kj}} \quad (5.2.4)$$

Donde N_{ij} es el número de veces de la palabra *i* ocurrida dentro de los documentos de la clase c_j en el conjunto de entrenamiento. Para evitar el problema de la probabilidad cero se utiliza Laplace (Se agrega el +1). M es el número de términos en el vocabulario.

A pesar de que la suposición de independencia condicional es generalmente falsa para la aparición de la palabra en documentos, el clasificador Naive Bayes es sorprendentemente efectivo.

5.3 Máquinas de Soporte Vectorial (SVMs)

Una Máquina de soporte Vectorial (SVM) genera la superficie decisión de dos clases distintas de los puntos de entrada. Como un clasificador de una sola clase, la descripción dada por los datos de los vectores de soporte es capaz de formar una frontera de decisión alrededor del dominio de los datos de aprendizaje con muy poco o ningún conocimiento de los datos fuera de esta frontera. Los datos son mapeados por medio de un kernel Gaussiano u otro tipo de kernel, el de un espacio de características, en un espacio dimensional más alto, donde se busca la máxima separación entre clases. Esta función de frontera, cuando es traída de regreso al espacio de entrada, puede separar los datos en todas las clases distintas, cada una formando un agrupamiento [5].

La teoría de las Máquinas de Soporte Vectorial (SVM por su nombre en inglés Support Vector Machines), es una nueva técnica de clasificación y ha tomado mucha atención en años recientes. La teoría de la SVM está basada en la idea de minimización de riego estructural (SRM) [3]. En muchas aplicaciones, las SVM han mostrado tener gran desempeño, más que las máquinas de aprendizaje tradicional como las redes neuronales y han sido introducidas como herramientas poderosas para resolver problemas de clasificación [5].

Una SVM primero mapea los puntos de entrada a un espacio de características de una dimensión mayor (si los puntos de entrada están en R^2 entonces son mapeados por la SVM a R^3) y encuentra un hiperplano que los separe y maximice el margen *m* entre las clases en este espacio [5].

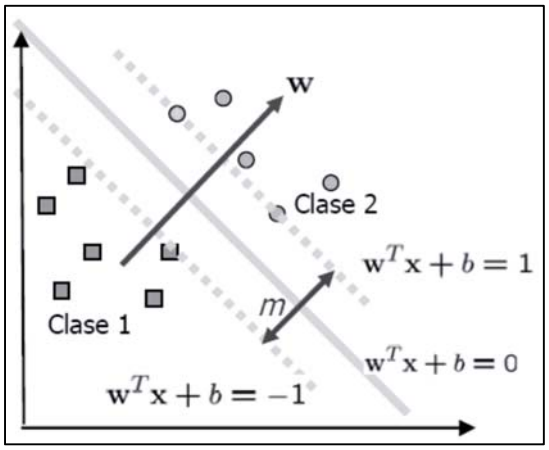

Figura 5.3: La frontera de decisión SVM.

Maximizar el margen m es un problema de programación cuadrática (QP) y puede ser resuelto por su problema dual introduciendo multiplicadores de Lagrange. Sin ningún conocimiento del mapeo, la SVM encuentra el hiperplano óptimo utilizando el producto punto con funciones en el espacio de características que son llamadas kernels. La solución del hiperplano óptimo puede ser escrita como la combinación de unos pocos puntos de entrada que son llamados vectores de soporte [5].

Actualmente, hay muchas aplicaciones que utilizan las técnicas de las SVM como por ejemplo las de OCR (Optical Character Recognition) por la facilidad de las SVMs de trabajar con imágenes como datos de entrada [5].

5.3.1 Caso linealmente separable

Supongamos que nos han dado un conjunto *S* de puntos etiquetados para entrenamiento [5].

$(x_1,y_1),\ldots,(x_i,y_i)$ (5.3.1.1)

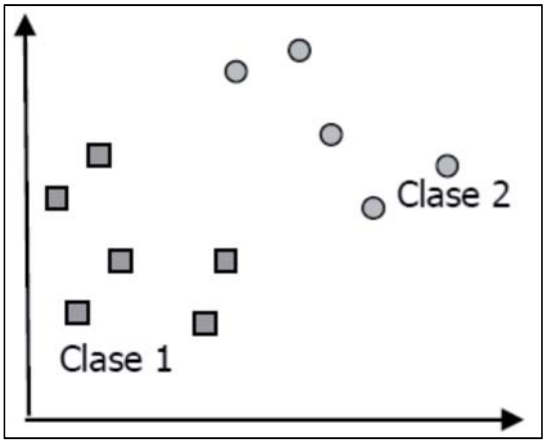

Figura 5.4 : Caso Linealmente Separable

Cada punto de entrenamiento $x_i \in R^N$ pertenece en alguna de dos clases y se le ha dado una etiqueta $y_i \in \{-1,1\}$ para $i = 1, \ldots, l$. En la mayoría de los casos, la búsqueda de un hiperplano adecuado en un espacio de entrada es demasiado restrictivo para ser de uso práctico. Una solución a esta situación es mapear el espacio de entrada en un espacio de características de una dimensión mayor y buscar el hiperplano óptimo allí. Sea $z=\varphi(x)$ la notación del correspondiente vector en el espacio de características con un mapeo φ de R^N a un espacio de características Z. Deseamos encontrar el hiperplano [5].

$$w \cdot z + b = 0 \quad (5.3.1.2)$$

Definido por el par (w, b), tal que podamos separar el punto x_i de acuerdo a la función.

$$f(x_i) = sign(w \cdot z_i + b) = \begin{cases} 1 & y_i = 1 \\ -1 & y_i = -1 \end{cases} (5.3.1.3)$$

Donde $w \in Z$ y $b \in R$. Más precisamente, el conjunto S se dice que es linealmente separable si existe (w, b) tal que las inecuaciones;

$$\begin{cases} (w \cdot z_i + b) \geq 1 & y_i = 1 \\ (w \cdot z_i + b) \leq -1 & y_i = -1 \end{cases} i = 1,\ldots..l \ (5.3.1.4)$$

Sean válidas para todos los elementos del conjunto *S*. Para el caso linealmente separable de *S*, podemos encontrar un único hiperplano óptimo, para el cual, el margen entre las proyecciones de los puntos de entrenamiento de dos diferentes clases es maximizado [5].

5.3.2 Caso No linealmente separable

Si el conjunto *S* no es linealmente separable, violaciones a la clasificación deben ser permitidas en la formulación de la SVM.

Figura 5.5 : Caso No Linealmente Separable

Para tratar con datos que no son linealmente separables, el análisis previo puede ser generalizado introduciendo algunas variables no-negativas de tal modo que (5.3.1.4) es modificado a;

$$y_i(w \cdot z_i + \mathrm{b}) \geq 1 - \varepsilon_i (5.3.1.5)$$

Los $\varepsilon_i \neq 0$ en la ecuación 2.5 son aquellos para los cuales el punto x_i no satisface la ecuación 2.4. Entonces el término $\sum_{i=1}^{l} \varepsilon_i$ puede ser tomado como algún tipo de medida del error en la clasificación[5].

El problema del hiperplano óptimo, es entonces redefinido como la solución al problema;

$$\min\{\frac{1}{2} w \cdot w + C \sum_{i=1}^{l} \varepsilon_i\}$$

$$y_i(w \cdot z_i + b) \geq 1 - \varepsilon_i, \quad i = 1, \ldots, l (5.3.1.6)$$

$$\varepsilon_i \geq 0, i = 1, \ldots, l$$

Donde C es una constante. El parámetro C puede ser definido como un parámetro de regularización. Este es el único parámetro libre de ser ajustado en la formulación de la SVM. El ajuste de éste parámetro puede hacer un balance entre la maximización del margen y la violación a la clasificación [5].

Buscando el hiperplano óptimo en (5.2.3.6) es un problema QP, que puede ser resuelto construyendo un Lagrangiano y transformándolo en el dual;

$$Max\ W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j z_i \cdot z_j \quad (5.3.1.7)$$

$$\text{Sea,} \quad \sum_{i=1}^{l} y_i\, \alpha_i = 0 \qquad 0 \le \alpha_i \le C, i = 1, \dots, l$$

Donde $\alpha = (\alpha_{1,\dots,}\alpha_l)$ es un vector de multiplicadores de Lagrange positivos asociados con las constantes en (5.3.2.5).

El teorema de Khun-Tucker juega un papel importante en la teoría de las SVM. De acuerdo a este teorema la solución para el vector α del problema satisface:

$$\alpha(y_i(w \cdot z_i + b) - 1 + \varepsilon_i) = 0, \quad i = 1, \dots, l \quad (5.3.1.8)$$

$$(C - \alpha)\varepsilon_i = 0, \quad i = 1, \dots, l \quad (5.3.1.9)$$

Para esta igualdad se deduce que para los únicos valores de $\alpha \neq 0$ (5.3.1.9) son aquellos que para las constantes en (5.3.1.5) son satisfechas con el signo de igualdad. El punto x_i correspondiente con $\alpha > 0$ es llamado vector de soporte. Pero hay dos tipos de vectores de soporte en un caso no separable. En el caso $0 < \alpha < C$, el vector de soporte x_i satisface las igualdades $\left(y_i(w \cdot z_i + b)\right) = 1\ y\ \varepsilon_i = 0$ en el caso de $\alpha = C$, el correpondiente ε_i es diferente de cero y el correspondiendo vector de soporte no satisface (2.4). Nos referimos a estos vectores de soporte como errores. El punto x_i correspondiente con $\alpha = 0$ es clasificado correctamente alejado del margen de decisión[5].

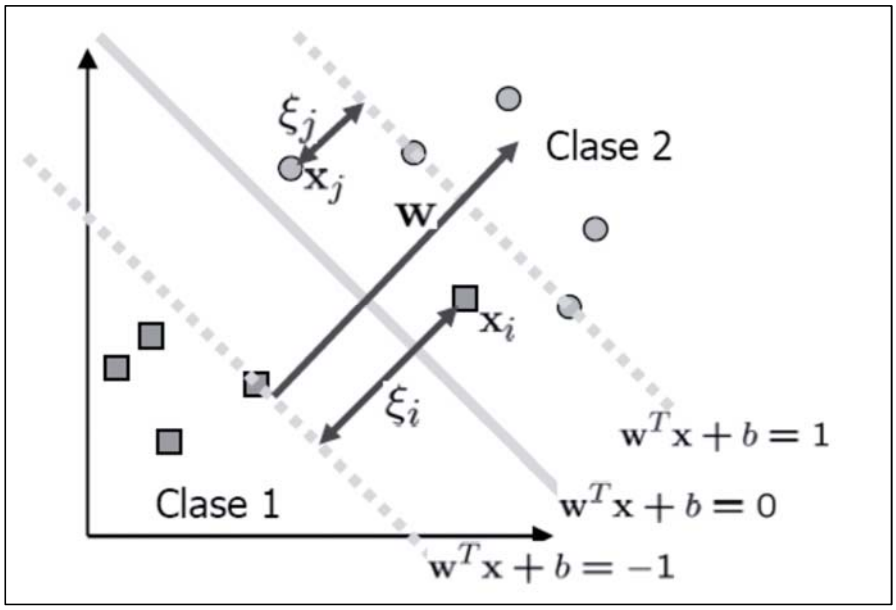

Figura 5.6: Aparición del parámetro error en clasificación.

5.3.3 Sequential Minimal Optimization - SMO

SMO es un algoritmo que puede resolver rápidamente la SVM sobre problemas cuadráticos (QP), sin necesidad de una matriz de almacenamiento y sin numerales cuadráticos para optimizar los pasos. SMO descompone en general el QP en sub-problemas cuadráticos, usando el teorema de Osuna's para asegurar la convergencia [6].

A diferencia de otros métodos, SMO elige resolver el problema de optimización más pequeño posible en cada paso. Para el problema estándar SVM QP, el problema de optimización más pequeño posible implica dos multiplicadores de Lagrange, debido a que los multiplicadores de Lagrange deben obedecer a una igualdad lineal [6].

La ventaja de SMO reside en el hecho de que la solución durante dos multiplicadores de Lagrange se puede hacer analíticamente. Por lo tanto, la optimización numérica QP se evita por completo. El bucle interior del algoritmo se puede expresar en una corta cantidad de código, en vez de invocar una biblioteca entera de QP. A pesar que con lleva a más iteraciones para resolver cada sub-QP es tan rápido que el problema global se resuelve rápidamente [6].

6. Representación del Problema

El rendimiento de un sistema de razonamiento, depende en gran medida en la representación del problema. La misma tarea puede ser fácil o difícil, dependiendo de la forma en que se describe.

La representación explicita de la información mejora los rendimientos de las máquinas. Además, una representación quizás más compleja o más desarrollada puede mejorar el funcionamiento de los algoritmos.

La representación de documentos tiene alto impacto en la tarea de clasificación. Algunos elementos incluyen Diagramas, palabras, frases o términos lógicos. Dentro de estos conjuntos la herramienta más utilizada para describir estos elementos es el modelo espacio vectorial, debido a su simplicidad conceptual y el atractivo de su metáfora subyacente de la utilización del espacio.

Con el vector de espacio vectorial (*vector space model* VSM), el contenido de un documento es representado mediante las componentes de espacio $d = \{w_1;\ldots;w_k\}$ donde *k* es el largo del conjunto. Los términos pueden ser medidos en varios niveles, tales como silabas, palabras, frases o cualquier otra unidad semántica. Diferentes términos tienen una importancia distinta en el texto y por lo tanto, la componente w_i indica la importancia del término (usualmente entre los valores de 0 y 1) t_i en la semántica del documento [9].

Uno de los métodos más eficaces para la ponderación de los pesos es la frecuencia de los términos en el documento [9].

Hay diferentes asignaciones de texto para el espacio de entrada a través de los clasificadores de texto. Leopold y Kindeman proponen, por ejemplo, combinaciones con diferentes asignaciones de funciones del kernel en máquinas de soporte vectorial. De acuerdo con Lan hay dos decisiones importantes para la elección de una representación basada en VSM. En primer lugar, ¿qué debería constituir un término? Por ejemplo, debería ser una sub-palabra, palabra o varias palabras con significado. En segundo lugar, ¿Cómo debería ser el término ponderado? La ponderación puede ser una función binaria o termino de frecuencia inversa en el documento (Frequency – Inverse Document Frequency tf - idf), desarrollado por Salton y Buckley, utilizando la función de selección métrica como X^2 ,Information Gain (IG) o Gain Ratio (GR). Los métodos de ponderación de términos mejoran la eficacia de la clasificación de texto mediante la asignación adecuada de valores. Aunque la clasificación de texto ha sido estudiada por varias décadas, los métodos de plazo de ponderación para la clasificación de texto son por lo general prestados de la recuperación de información (Information Retrieval IR) [9].

Tabla 6.1: Representación de los términos con las etiquetas y pesos

	T	***t'***
Etiquta$_1$	a_t,λ_1	d_t,λ_1
Etiquta$_2$	a_t,λ_j	d_t,λ_j
Etiquta$_3$	$a_t,\|L\|$	$d_t,\|L\|$

6.1 Representación con Conjuntos de Palabras (TF-IDF)

La representación de documentos más utilizada para la clasificación de texto es tf-idf, donde un problema con dos clases (Etiqueta 1 representa la clase+ y la etiqueta 2 representa la clase-) en donde cada componente del vector se calculó como[9]:

$$tf-idf_{td} = f_{t,d} \log_{10}\left(\frac{N}{N_t}\right) \quad (6.1)$$

Donde $f_{t,d}$ es la frecuencia del término t en el documento d, N es igual a la expresión: $(a_{t,\lambda_1} + d_{t,\lambda_1} + a_{t,\lambda_2} + d_{t,\lambda_2})$ que corresponde al número de documentos, y N_t es igual a $(a_{t,\lambda_1} + a_{t,\lambda_2})$ donde es el número de documentos que contienen el termino t[9].

6.2 Representación de Frecuencia Relevante (TF-RF)

Lan propuso *tf-rf* como una representación basada en la mejora SVM usando dos clases y los problemas de etiqueta única (donde, Etiqueta 1 representa la clase+ y la etiqueta 2 representa la clase-) [9].

$$tf-rf_{td} = f_{t,d} \log_2\left(2 + \frac{a_{t,\lambda_1}}{\max\left(1, a_{t,\lambda_2}\right)}\right) \quad (6.2)$$

Dónde:

- $f_{t,d}$ es la frecuencia del término en el documento d.
- a_{t,λ_1} es el número de documentos en la clase positiva que contienen el término *t*.

- a_{t,λ_2} es el número de documentos en la clase negativa que contienen el término *t.*
- $\max(1, a_{t,\lambda_2})$ función en el denominador que permite que el término no se indefina.

6.3 Representación del Problema con Etiquetas Múltiples

Tf-idf representa los documentos considerando solamente la frecuencia de términos en el documento (*tf*), y la frecuencia de términos en todos los documentos (*idf*), sin tomar en cuenta que las clases y etiquetas pertenecen a los documentos. Por otra parte, *tf-rf* también considera la frecuencia de términos en el documento (*tf*) y la frecuencia de los términos en todos los documentos bajo la clase de evaluación (*rf*). En *tf-rf,* cada documento es representado por un vector diferente cuando es evaluado en una clase particular. Desde un punto de vista teórico, esta extensión de *tf-rf* representa los cambios del texto que se está representando en el documento, de acuerdo a una etiqueta en evaluación, logrando grandes diferencias entre los documentos pertenecientes a diferentes etiquetas y aprovechando el desempeño de los clasificadores binarios. Así la información importante acerca de la frecuencia en otras clases se puede utilizar [9].

En este caso se propone usar una funcionalidad de centralidad μ-Frecuencia Relevante de una etiqueta (tf-μrfl). La nueva representación donde la representación del documento de visualiza como un vector de frecuencias. Esta nueva forma constituye una nueva representación a partir de la *tf-rf* para un problema ML [9].

Tabla 6.2: Ejemplo de Frecuencia de un término por cada etiqueta

	E_1	**E_2**	**E_3**	**E_4**	**E_5**	**E_6**	**E_7**	**E_8**	**E_9**
Frecuencia	53	76	87	55	66	45	32	54	23

(6.3) $$tf{-}\mu rfl_{tdl} = f_{t,d}\log_2\left(2 + \frac{a_{t,l}}{\mu(a_{t,\lambda_{j/l}})}\right)$$

Donde $\mu(a_{t,\lambda_{j/l}})$ es la función sobre el conjunto de datos $a_{t,\lambda_{j/l}} = \{a_{t,\lambda_1}, ..., a_{t,\lambda_{l-1}}, a_{t,\lambda_{l+1}}, ..., a_{t,|L|}\}$. Se considera además $\mu(a_{t,\lambda_{j/l}}) = \max(1, mean(a_{t,\lambda_{j/l}}))$ para representar *tf-rfl*y

$\mu(a_{t,\lambda_{j/l}}) = \max\left(1, median(a_{t,\lambda_{j/l}})\right)$ para la representación de *tf-rrfl*. Cada función entrega medidas de centralidad, con una medida clásica y mediana robusta [9].

6.3.1 Frecuencia de Etiqueta Relevante(TF-RFL)

La frecuencia de etiqueta relevante (*tf-rfl*) es derivada desde la *tf-μrfl* y constituye una nueva representación para un problema de múltiple etiqueta.

$$tf{-}rfl_{tdl} = f_{t,d}\log_2\left(2 + \frac{a_{t,l}}{\max\left(1, mean(a_{t,\lambda_{j/l}})\right)}\right) \quad (6.3.1)$$

En la ecuación, el término $mean(a_{t,\lambda_{j/l}})$ es el promedio del número de documentos que contienen el término t por cada documento etiquetado que no sea *l*[9].

6.3.2 Frecuencia de Etiqueta Relevante Robusta(TF-RRFL)

La frecuencia de etiqueta relevante robusta (*tf-rrfl*), es derivada desde la μ-frecuencia relevante de etiqueta (*tf-μrfl*) esta es una segunda representación para los problemas de múltiple etiqueta

$$tf{-}rrfl_{tdl} = f_{t,d}\log_2\left(2 + \frac{a_{t,l}}{\max\left(1, median(a_{t,\lambda_{j/l}})\right)}\right) \quad (6.3.2)$$

El uso de la media, es más robusta en los resultados de los conjuntos de datos que contiene grandes diferencias entre la frecuencia en que aparece el término en un conjunto de etiquetas bajo otras etiquetas en evaluación [9].

7. Datos de Etiquetas Múltiples

En esta sección se explicará la forma de medir como están dispersas las etiquetas en los datos y que variables se miden.

7.1 Medición de Etiquetas Múltiples

Las etiquetas simples y etiquetas múltiples, pueden ser medidas por el número de ejemplos(N), el número de atributos del espacio de entrada (M) o el número de etiquetas (L). En esta sección revisaremos las medidas específicas para las etiquetas múltiples.

7.1.1 Label Cardinality

Label Cardinality – (LCard) (ecuación 7.11) es la medición estándar de "multi-labelled-ness", introducido por Tsoumakas y Katakis en el 2007. Esto es simplemente el promedio de etiquetas asociadas por cada ejemplo.

$$\mathrm{LC}_{\mathrm{ARD}}(D) = \frac{\sum_{i=1}^{N} |y_i|}{N} \quad (7.1.1)$$

7.1.2 Label Density

Label Density (LDens) (ecuación 7.1.2), también introducido por Tsoumakas y Katakis en el 2007, depende directamente de LCard, sin embargo, esta medida toma en cuenta el tamaño del espacio de la etiqueta.

$$\mathrm{LD}_{\mathrm{ENS}}(D) = \frac{1}{L}\mathrm{LC}_{ARD}(D) \quad (7.1.2)$$

7.1.3 Proportion of Unique y Proportioni of Ocurrences

A diferencias de las medidas anteriores, las medidas de Proportion of Unique y Proportion of Ocurrences tienen medición sobre la regularidad o uniformidad del conjunto de etiquetas. Proportion of Unique (ecuación 7.1.3) realiza combinación de etiquetas y Proportion of Ocurrences (ecuación 7.1.4) representa la más alta frecuencia que ocurre en las etiquetas.

$$\mathrm{PU}_{NIQ}(D) = \frac{|\{y \mid \exists! x : (x,y) \in D\}|}{N} \quad (7.1.3)$$

$$\mathrm{PM}_{AX}(D) = \max_{y|(x,y)\in D} \frac{\mathrm{COUND}(y,\mathrm{D})}{N} \quad (7.1.4)$$

Una Alta *PUniq*(D) indica una irregularidad de las etiquetas, y cuando existe una alta PMax(D) significa que las etiquetas exhiben labelskew antes conocido como powerlaw. En el contexto de las etiquetas multiples label skew traduce la relación que existe entre el número más alto ejemplos asociados, con la etiqueta más común del conjunto.

7.2 Medidas del Dataset

Como hemos visto en el capítulo 4 se expusieron las ecuación o fórmulas para calcular las medidas de las etiquetas del dataset, en esta sección se utilizarán las fórmulas en los datos de la investigación.

7.2.1 Variables del Dataset

Existen un gran número de variables que se pueden extraer del dataset pero todas están depende de tres variables:

1. El número de ejemplos o instancias de un elemento, se expresa con la letra N.
2. El número de etiquetas del dataset, se expresa con la letra L.
3. El número de palabras que existen en el dataset corresponde a la letra M.

En los datos de la investigación se tiene:

Tabla 7.1: Variables del dataset de registros clínicos.

N	L	M	LCard	LDens
49979	2889	24114	1.2891440973	0,00044623

7.3 Distribución de las Etiquetas

La distribución de las etiquetas es el análisis como se distribuyen las frecuencias de las mismas en la información. Podemos realizar este análisis con las fórmulas o ecuación que se expusieron en la sección 4.1.

Un LCard cercano a 1.0 indica la mayoría de los ejemplos o documentos asociados a una sola etiqueta.

LDens usualmente es muy baja esto indica que los documentos están muy esparcidos entre las etiquetas del espacio escogido.

8. Estadística de las Historias Clínicas

Para este trabajo hemos tomado las historias clínicas y consultas dentro de un período de 6 años del sistema de un sistema HIS.

A continuación, se entrega la variabilidad de los diagnósticos clínicos registrados.

Dentro de este período se han filtrado 49977 registros de atención, donde el largo del registro contiene más de 100 caracteres. Donde se han se han registrado 2889 diagnósticos distintos.

La media de los diagnósticos por la cantidad de registro es:

$$Media = \frac{49977}{2889} = 17{,}299065$$

La varianza de los diagnósticos:

$$Varianza = \ 11027{,}96385$$

La desviación estándar de los diagnósticos

$$Desviación = 105{,}014113$$

En la siguiente tabla se agrupan los diagnósticos según la frecuencia de uso. El intervalo de 1 a 500 indica que hay 2872 diagnósticos con ese intervalo de repetición o frecuencias. Así, por ejemplo, en el intervalo 3501 a 4000 hay solo un diagnóstico con tal repetición.

Tabla 8.1 : Agrupación de Frecuencias de diagnósticos.

INTERVALOS		FRECUENCIAS	FRECUENCIA ACUM.	FRECUENCIA ACUM %
1	500	2872	2872	99,41%
501	1000	9	2881	99,72%
1001	1500	5	2886	99,90%
1501	2000	2	2888	99,97%
2001	2500	0	2888	99,97%
2501	3000	0	2888	99,97%
3001	3500	0	2888	99,97%
3501	4000	1	2889	100,00%

Los registros utilizados están muy dispersos y crecen rápidamente hasta llegar a una cantidad de 271 diagnósticos, esto equivale a casi el 10% del total de diagnósticos utilizados en la muestra de datos. Por otro lado, este 10% de diagnósticos cubre el 80% de los registros.

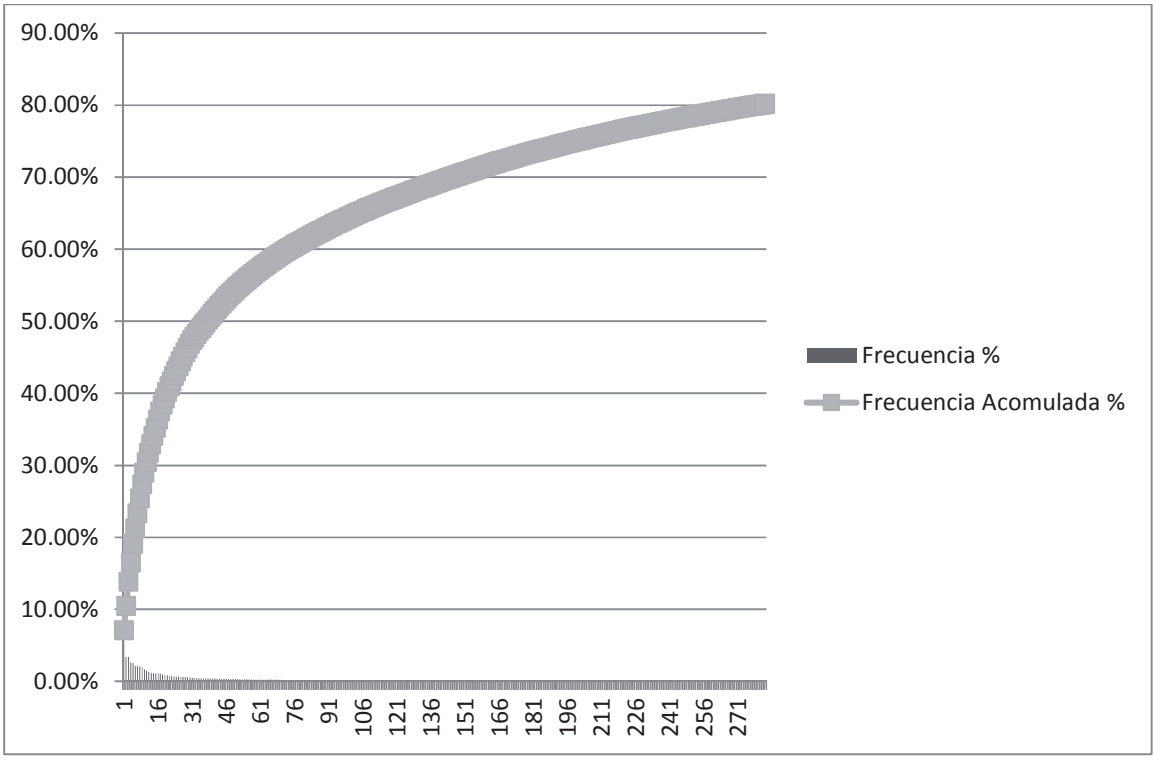

Figura 8.1: Gráfico Pareto Diagnósticos.

9. Evaluación

La evaluación de los algoritmos de aprendizaje, corresponden a las métricas capaces de dar información relevante al desempeño de la clasificación. A continuación, se presentan las métricas para la evaluación.

9.1 Matriz de Confusión

En el campo de la inteligencia artificial, una matriz de confusión es una herramienta visual que se utiliza en el aprendizaje supervisado, cada columna que posee representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias de la clase real. Uno de los principales beneficios de las matrices de confusión es que facilitan ver si el sistema se confunde entre dos clases [4].

		CLASIFICADO	
		POSITIVO	NEGATIVO
REAL	POSITIVO	Verdadero Positivo (VP)	Falso Negativo (FN)
	NEGATIVO	Falso Positivo (FP)	Verdadero Negativo (VN)

A partir de la matriz de confusión es que se pueden obtener métricas para la evaluación del clasificador.

9.2 Sensibilidad (Recall)

La sensibilidad indica la capacidad del estimador para dar como casos positivos los casos que realmente lo son. La sensibilidad viene ser la razón de los verdaderos positivos [2].

Para calcularse se utiliza:

$$recall = \frac{VP}{VP + FN}$$

9.3 Precisión

La precisión es el cociente entre los verdaderos positivos y la suma de los verdaderos positivos y los falsos positivos, como se muestra a continuación:

$$Precisión = \frac{VP}{VP + FP}$$

9.4 Valor F

El valor F es la medida de la precisión que tiene una clasificación y utiliza los valores obtenidos de la *precisión y recall* [3].

Para calcularse se utiliza:

$$F_1 = 2 * \frac{Precisión * Recall}{Precisión + Recall}$$

10. Prototipo Implementado

Para realizar el método de CHAIN con votaciones a través de frecuencias se desarrolló un sistema en lenguaje Java, a continuación, se explicará cada parte del modelo.

10.1 Prototipo

El prototipo planteado utiliza una representación vectorial de los documentos, usando un diccionario de datos. Cada documento (historia clínica), se transforma en un vector y cada palabra se codifica según el diccionario. El dominio del vector corresponde al diccionario de palabras conocidas para el clasificador (N).

Una vez representado los documentos, el proceso para construir cada clasificador se basa en el algoritmo de CHAIN, calculando además las frecuencias.

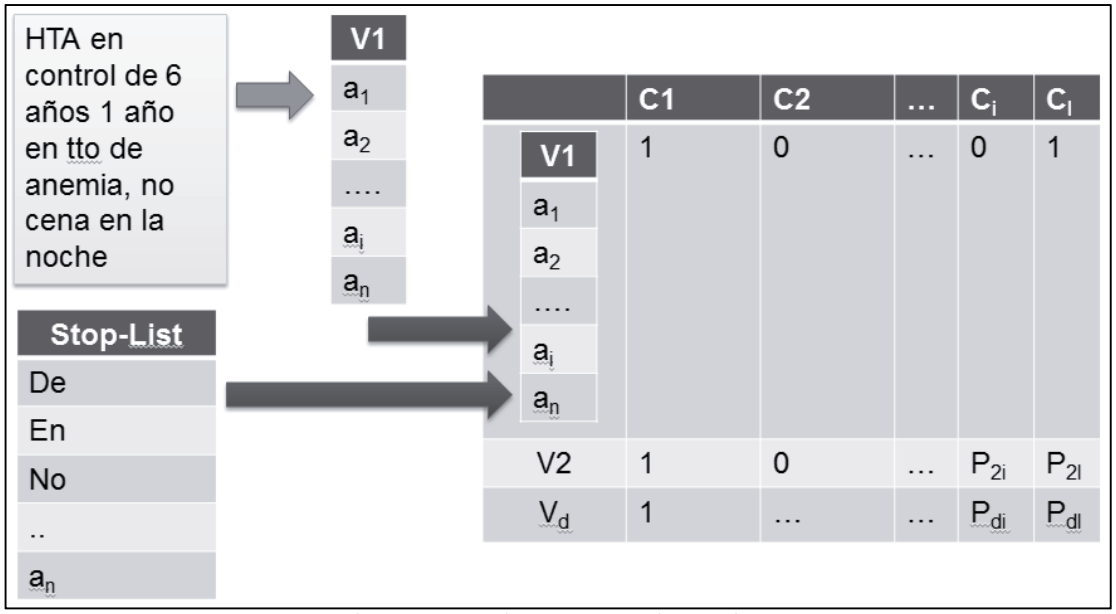

Figura 10.1 : Primer Paso en el Prototipo.

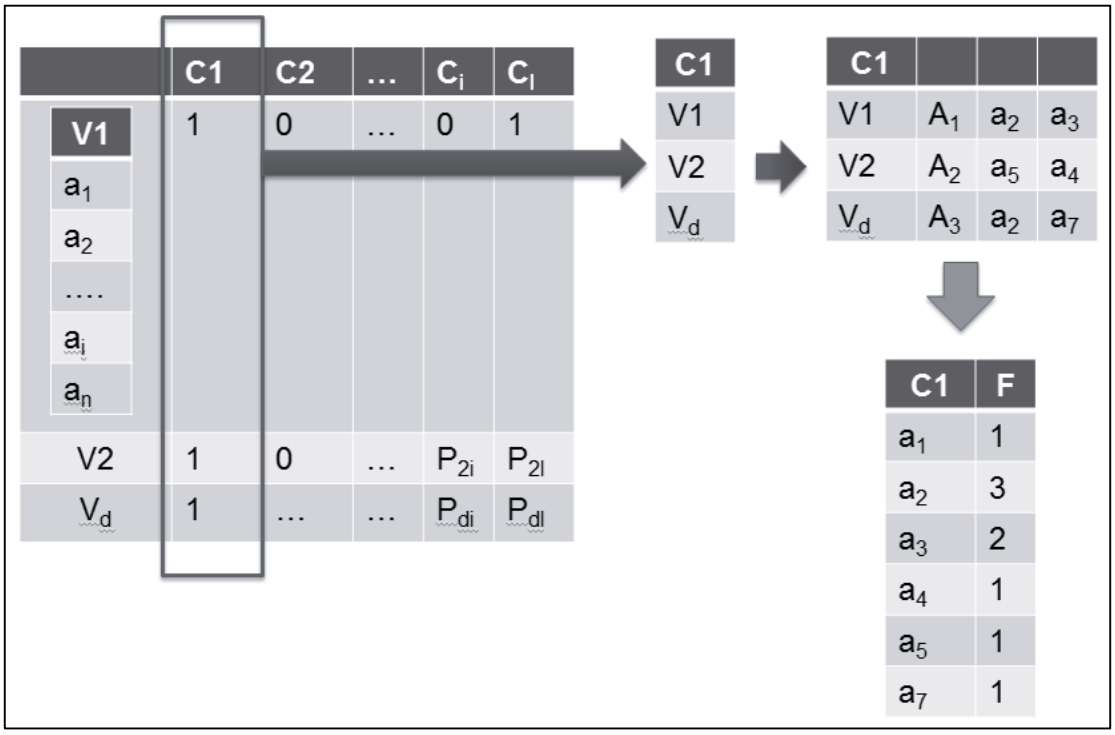

Figura 10.2 : Segundo Paso en el Prototipo.

El proceso de clasificar se realiza a través de máquinas vectoriales, se utiliza el clasificador entregado por el algoritmo de CHAIN para calcular los pesos (w) en la ecuación 5.3.1.2, que son representados por las frecuencias calculadas en el algoritmo de CHAIN. Queda por calcular el parámetro b de la ecuación5.3.1.2. Este parámetro se calcula por cada clasificador según *tf – idf* ver ecuación 6.1, para todos los términos de un clasificador, calculando la media de la frecuencia.

11. Resultados de Pruebas

Para la evaluación de los clasificadores se utilizaron las tres métricas ya mencionadas previamente; precisión, sensibilidad (Recall) y Valor-F. Los Valores numéricos en de los gráficos pueden ser encontrados en el anexo A de este trabajo, y en el Anexo E, F y G se encuentra la tabla de resultados de los algoritmos.

Para las pruebas se utilizó el 60% de los registros para generar el modelo y un 40% para generar las pruebas.

11.1 Hipótesis 1

Para el primer caso se utiliza solo la historia clínica para determinar la categoría o diagnóstico correspondiente.

11.1.1 Resultados de la Precisión en Hipótesis 1.

En los siguientes gráficos se muestra la precisión que se obtuvo al utilizar solo la historia clínica en los algoritmos.

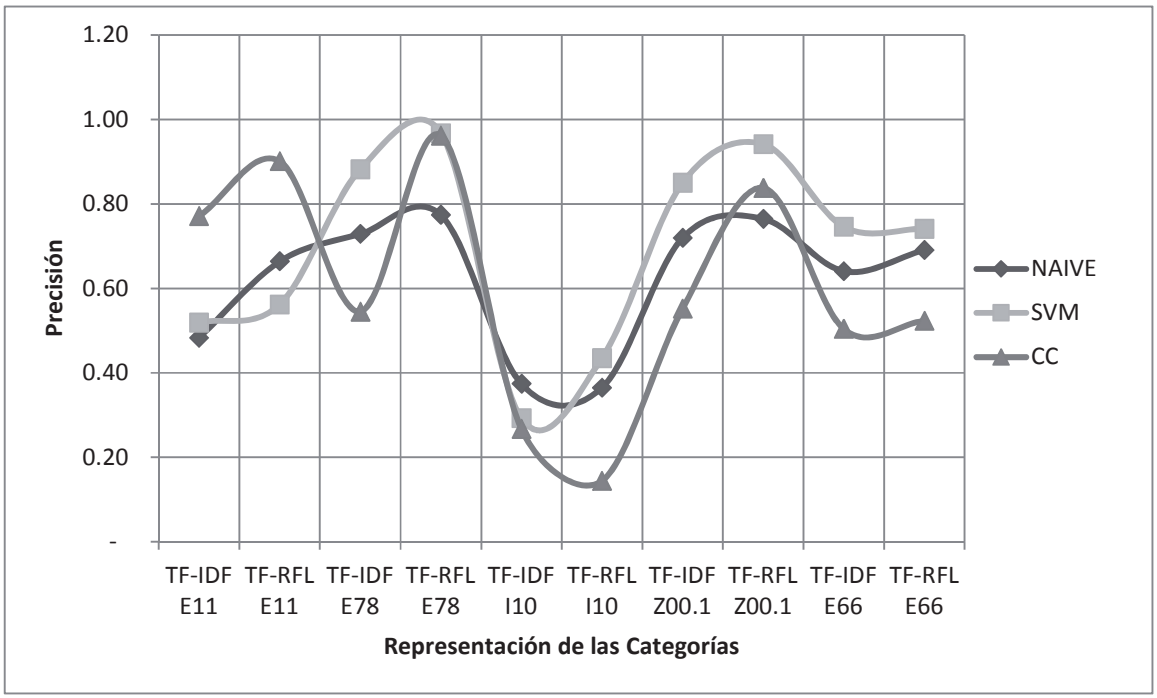

Figura 11.1: Precisión para las categorías en Hipótesis 1.

En los resultados de la precisión para la hipótesis 1 se visualiza en casi todas las categorías una supremacía de la SVM sobre los demás algoritmos. Solo para la categoría E11 el clasificador cadena superó a los otros dos algoritmos.

11.1.2 Resultados de la Sensibilidad (Recall) en Hipótesis 1.

En los siguientes gráficos se muestra la sensibilidad que se obtuvo al utilizar solo la historia clínica en los algoritmos.

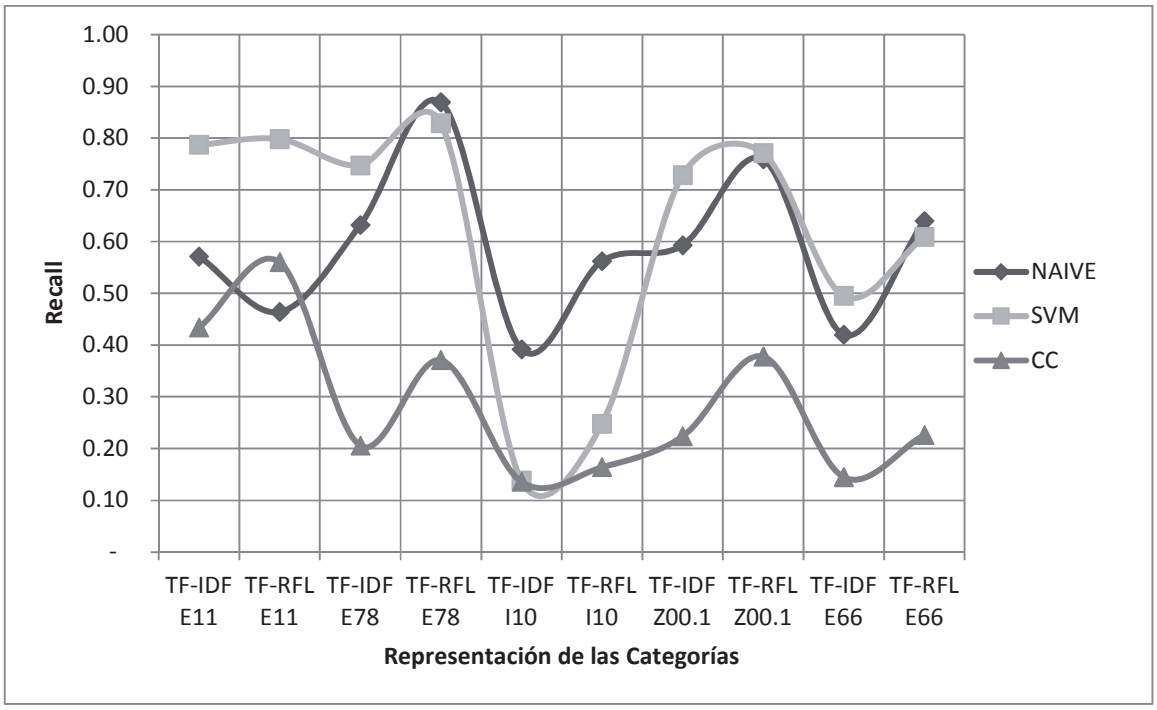

Figura 11.2 : Sensibilidad para las categorías en Hipótesis 1.

En los resultados de sensibilidad, queda claro que el algoritmo cadena tiene una baja sensibilidad detrás de los demás algoritmos.

11.1.3 Resultados del Valor F en Hipótesis 1.

En los siguientes gráficos se muestra el Valor F que se obtuvo al utilizar solo la historia clínica en los algoritmos.

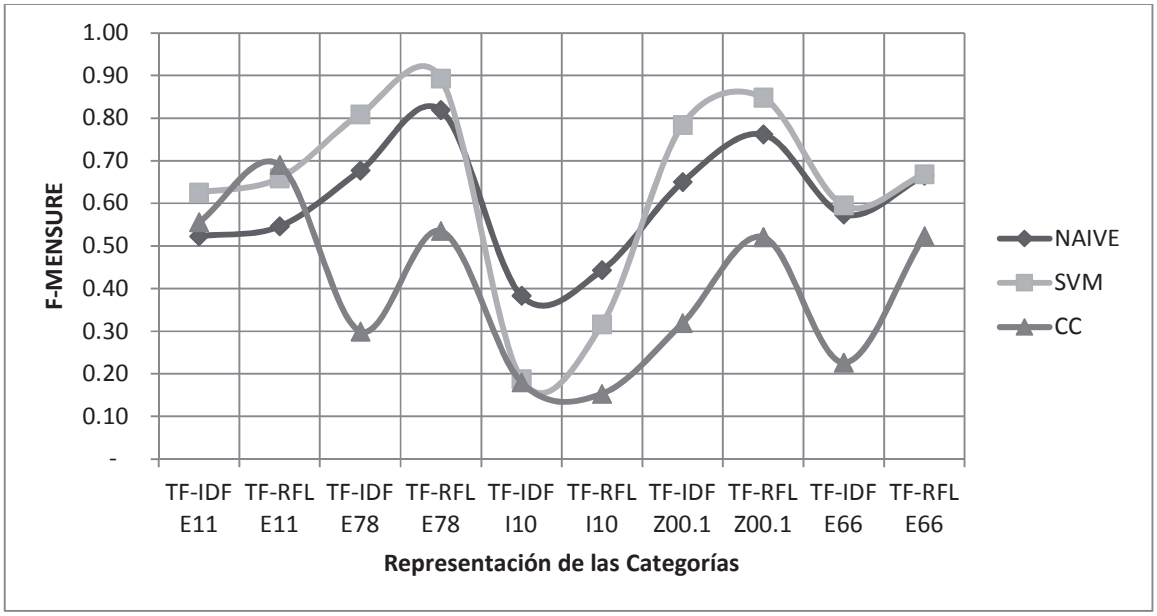

Figura 11.3 : Valor F para las categorías en Hipótesis 1.

En los resultados del valor F para la hipótesis 1 no hay clara una diferencia entre Naive Bayes y la SVM. El algoritmo cadena está en promedio bajo 0,2 puntos de los demás algoritmos.

11.1.4 Conclusión Hipótesis 1.

La SVM marca una diferencia en la clasificación para las categorías E78, E66 y Z00.1 pero la eficacia para la categoría I10 baja hasta el punto que es inferior que Naive Bayes y la Clasificación Cadena. El clasificador cadena obtuvo la mejor precisión para la categoría E11.

Además de los algoritmos se muestra que en todos los casos la representación TF-RFL fue superior que TF-IDF.

11.2 Hipótesis 2

Para el segundo caso, se utiliza solo la consulta del médico para determinar la categoría o diagnóstico correspondiente.

11.2.1 Resultados de la Precisión en Hipótesis 2.

En los siguientes gráficos se muestra la precisión que se obtuvo al utilizar solo la historia clínica en los algoritmos.

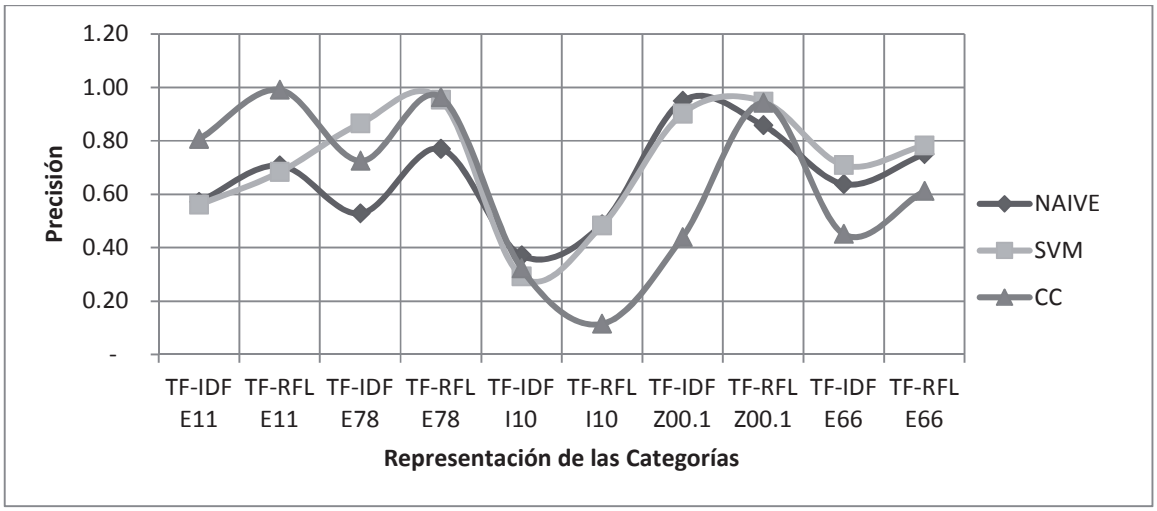

Figura 11.4: Precisión para las categorías en Hipótesis 2.

En los resultados de la precisión para la hipótesis 2se visualiza una gran similitud entre SMO y Naive Bayes, el algoritmo cadena queda un poco rezagado a los demás.

11.2.2 Resultados de la Sensibilidad (Recall) en Hipótesis 2.

En los siguientes gráficos se muestra la sensibilidad que se obtuvo al utilizar solo la consulta médica en los algoritmos.

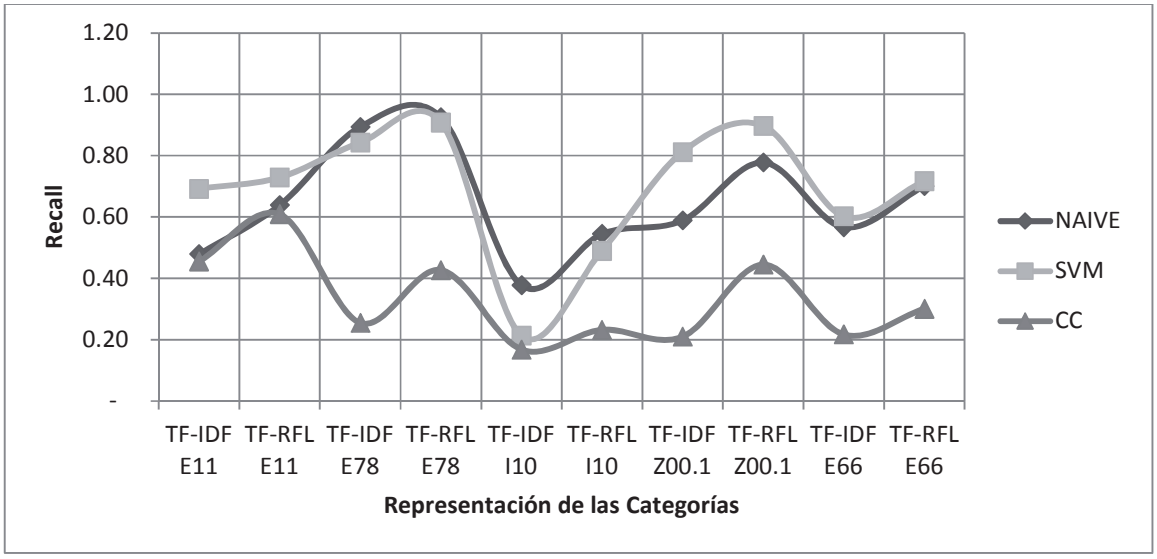

Figura 11.5 : Sensibilidad para las categorías en Hipótesis 2.

En los resultados de sensibilidad queda claro que el algoritmo cadena tiene una baja sensibilidad detrás de los demás algoritmos.

11.2.3 Resultados del Valor F en Hipótesis 2.

En los siguientes gráficos se muestra el Valor F que se obtuvo al utilizar solo la historia clínica en los algoritmos.

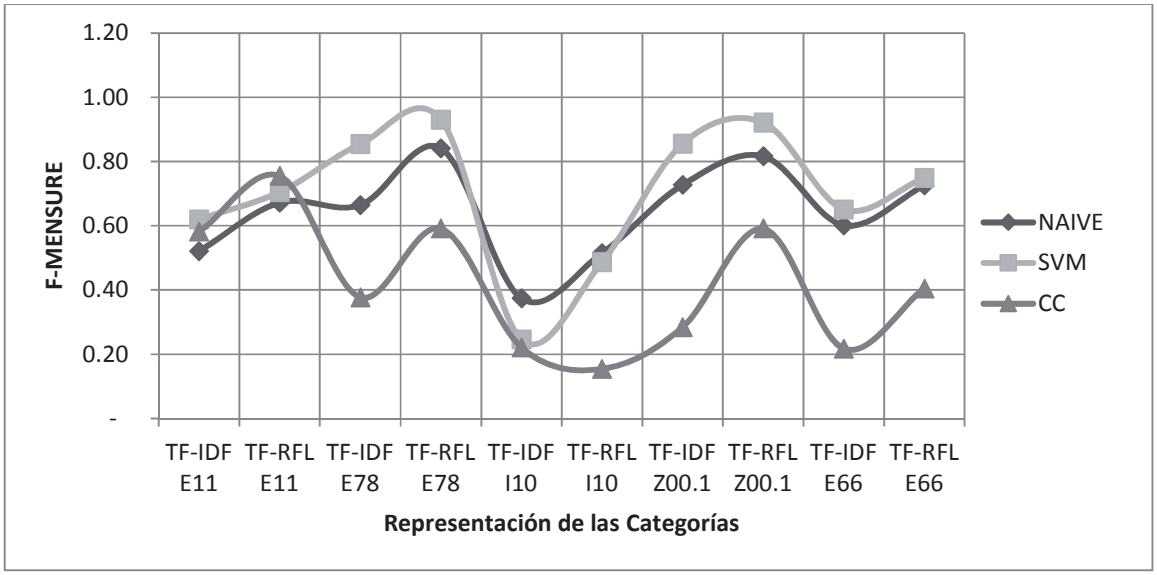

Figura 11.6 : Valor F para las categorías en Hipótesis 2.

En los resultados del valor F para la hipótesis 1, no hay clara una diferencia entre Naive Bayes y la SVM. El algoritmo cadena está en promedio bajo 0,2 puntos de los demás algoritmos.

11.2.4 Conclusión Hipótesis 2.

La SVM marca una diferencia en la clasificación para todas las categorías. La Clasificación Cadena y Naive Bayes están a la par menos en la categoría E78 donde Naive Bayes marca la diferencia.

11.3 Hipótesis 3

Para el tercer caso se utiliza la historia clínica y la consulta médica para determinar la categoría o diagnóstico correspondiente.

11.3.1 Resultados de la Precisión en Hipótesis 3.

En los siguientes gráficos se muestra la precisión que se obtuvo al utilizar la historia clínica y la consulta médica en los algoritmos.

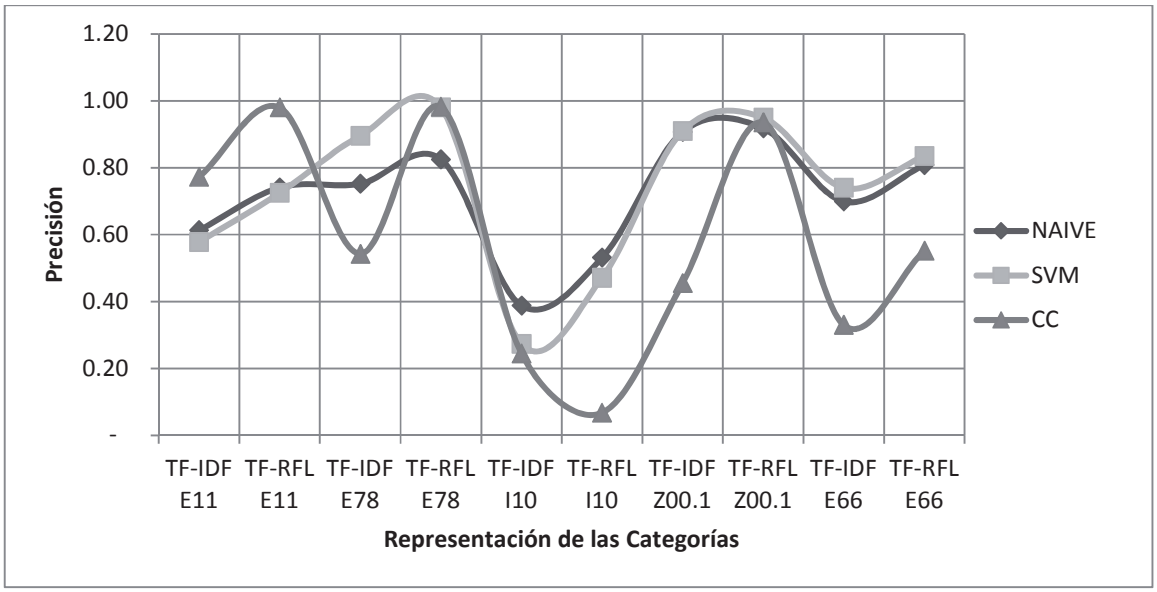

Figura 11.7: Precisión para las categorías en Hipótesis 3.

En los resultados de la precisión para la hipótesis 2, se visualiza una gran similitud entre SMO y Naive Bayes, el algoritmo cadena queda un poco rezagado a los demás.

11.3.2 Resultados de la Sensibilidad (Recall) en Hipótesis 3.

En los siguientes gráficos se muestra la sensibilidad que se obtuvo al utilizar solo la historia clínica en los algoritmos.

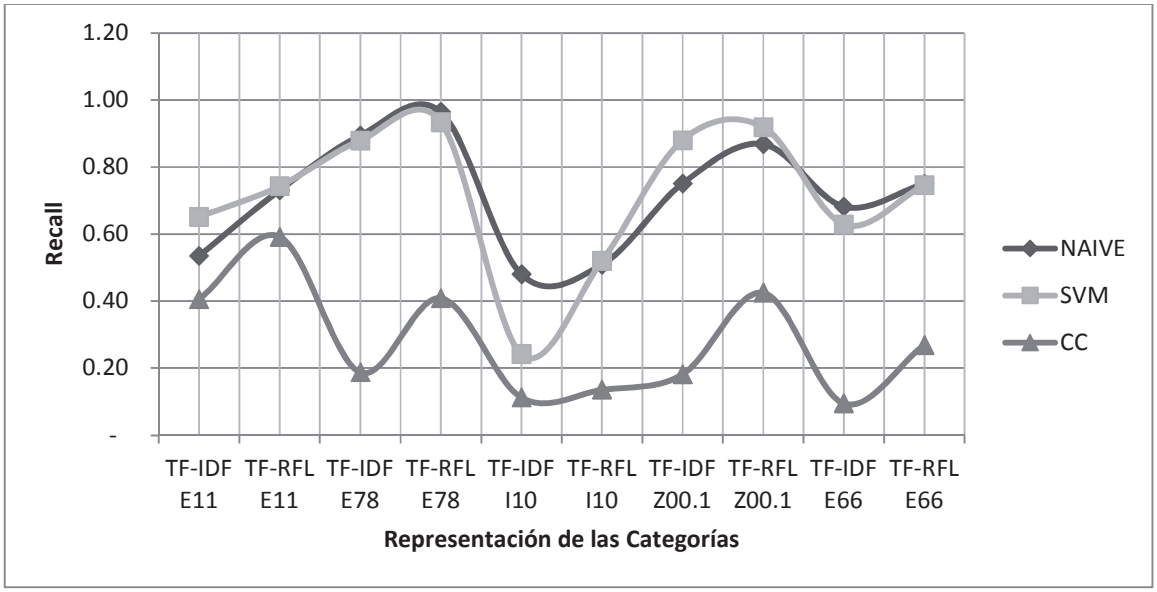

Figura 11.8 : Sensibilidad para las categorías en Hipótesis 3.

En los resultados de la precisión para la hipótesis 2, se visualiza una gran similitud entre SMO y Naive Bayes, el algoritmo cadena queda un poco rezagado a los demás.

11.3.3 Resultados del Valor F en Hipótesis 3.

En los siguientes gráficos se muestra el Valor F que se obtuvo al utilizar solo la historia clínica en los algoritmos.

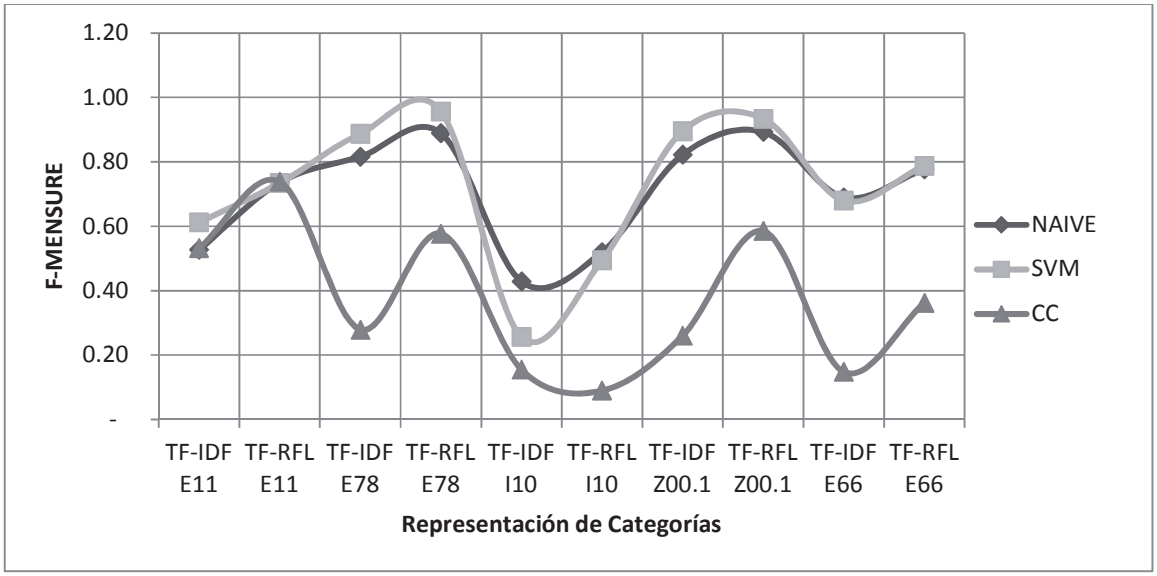

Figura 11.9 : Valor F para las categorías en Hipótesis 3.

En los resultados de la precisión para la hipótesis 2, se visualiza una gran similitud entre SMO y Naive Bayes, el algoritmo cadena queda un poco rezagado a los demás.

11.3.4 Conclusión Hipótesis 3.

Para esta prueba los tres algoritmos están muy cerca en la presión aun así SMO supera a Naive Bayes y el Algoritmo Cadena. La representación TF-RFL también fue la que obtuvo mejor resultados.

11.4 Rendimientos de Algoritmos

Para los algoritmos propuestos, se muestra en el gráfico los rendimientos promedio para las hipótesis.

El sistema utilizado fue Windows 7 con 8Gb de memoria RAM y un procesador Intel Core i5-3317 de 1.7GHz.

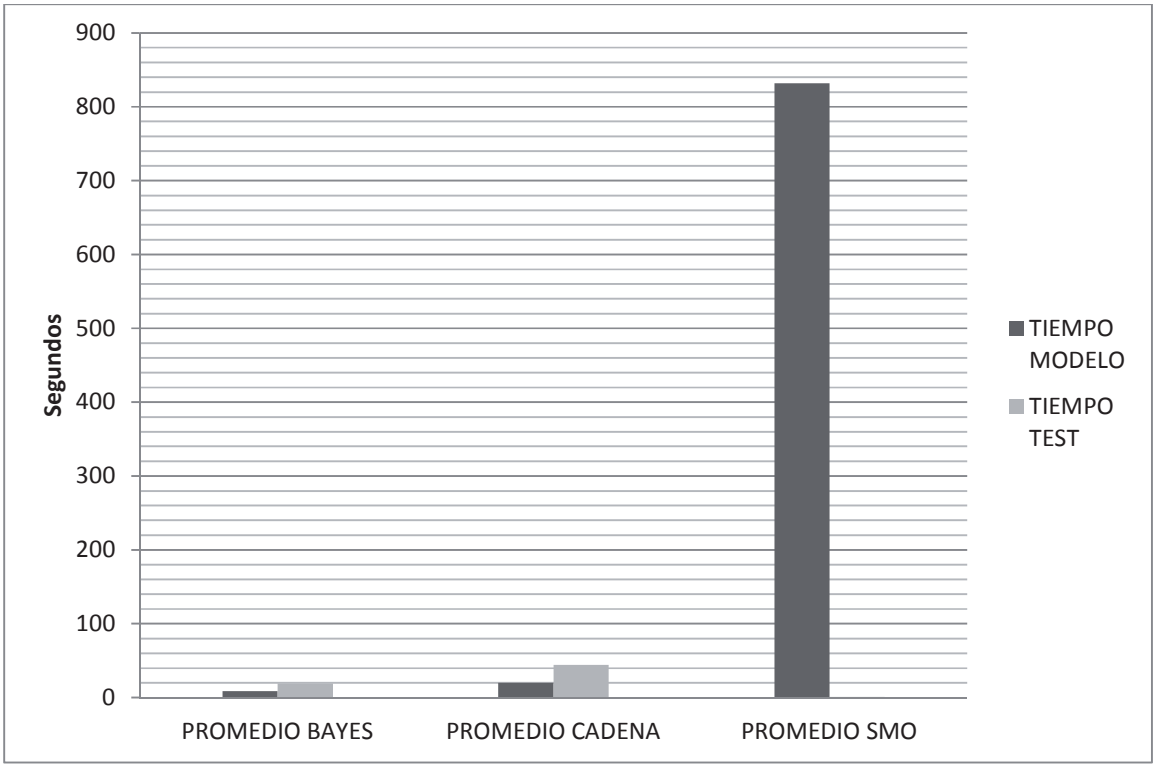

Figura 11.10: Promedio Rendimiento de Algoritmos

Para el caso de los algoritmos que se utilizaron para BAYES y SMO, se usaron una cantidad menor de parámetros puesto que con la cantidad de parámetros original la máquina nunca arrojó un resultado.

En el anexo H se encuentra la cantidad de parámetros utilizados para cada algoritmo. La condición que se utilizó para filtrar los parámetros fue el número de frecuencia por categoría, cantidad utilizada mínima 7.

Cabe destacar que el algoritmo SMO era instantáneo al realizar el test.

12. Conclusión.

La clasificación de texto es una rama de estudio que está continuamente en desarrollo y reinventándose con pequeños cambios, generando grandes impactos en la eficiencia y eficacias de los algoritmos.

Es complejo determinar si un contenido es óptimo de clasificar, ya que por la gran cantidad de técnicas que actualmente existen, quizás no es posible probar en todos, por características económicas o por otro tipo. Por lo tanto, se ha visto que en esta ciencia ya los algoritmos se asocian a contenidos, entonces no es necesario probar todo el universo, aun así pueden a ver excepciones que escapan a las estadísticas.

Las técnicas propuestas en este informe, por su clasificación son técnicas que deben ser supervisadas, y son generadas con datos de entrenamiento. Naive Bayes, SMO y Clasificación Cadena, todas plantean soluciones a los problemas de clasificación de texto enfocándose en no aumentar la complejidad del modelo y en generar un rápido resultado.

Además de los tipos de técnicas, se revisó la representación de los datos para poder resolver la clasificación y sin duda es una variable importante, que impacta directamente con la eficiencia y eficacia del clasificador. En este trabajo se compararon las técnicas TF-IDF y TF-RFL.

Las técnicas y representaciones estudiadas fueron aplicadas sobre datos clínicos, el historial del paciente, la consulta médica y el diagnóstico CIE-10.

Para los códigos CIE-10, existen más de diez mil diagnósticos y los datos propuestos, siendo filtrados previamente, estaban en el orden de cincuenta mil registros. Dentro de estos cincuenta mil registros, solo se visualizó dos mil ochocientos ochenta y nueve diagnósticos. En los datos utilizados se observó mucha dispersión, por la gran cantidad de diagnósticos que existen. Por lo tanto, para probar los clasificadores se escogieron los cinco diagnósticos con más frecuencia (I10, E11, E78, Z00.1 y E66).

El prototipo propuesto Clasificador Cadena, tiene ventajas en el rendimiento y además obtuvo una buena precisión en los resultados, aun así, no logró superar al SMO. Cabe destacar que siempre fue la mejor en estimar a la categoría E11 y con la hipótesis 2 usando solamente la consulta se mantuvo a los demás algoritmos. Además de eso, no fue necesario bajar la complejidad en el número de parámetros para poder generar las pruebas, el cual hace que el algoritmo se más confiable o tenga menos sensibilidad a los cambios.

Con Naive Bayes, se generó un clasificador rápidamente bordeando en casi todas las representaciones los diez segundos, pero luego las evaluaciones eran más costosas por el gran número de índices que tiene el vector de representación. Naive Bayes en casi todas sus pruebas superó al Clasificador Cadena y en algunos casos al algoritmo SMO.

El algoritmo SMO fue siempre superior en casi todas las pruebas que los demás algoritmos. Pero en la categoría I10 la cual para todos los algoritmos fue la peor evaluada SMO fue siempre el que obtuvo la peor evaluación. En el rendimiento SMO fue por lejos el que más tiempo demoró en construir el modelo el cual en promedio dura 800 segundos, pero fue el más rápido en testear los casos de prueba.

En las representaciones se obtiene que TF-RFL es mejor para estos casos que TF-IDF. Ya que se entiende que TF-RFL construye un modelo más preciso cuando se tienen varias categorías.

La categoría peor evaluada, es la categoría que tenía mayor frecuencia en la muestra utilizada. De un principio se pensó que esta categoría iba ser la mejor evaluada, pero luego de obtener las prueba se entiende que I10 por su gran número de registros captó mayor cantidad de Falsos Positivos por lo tanto arrastró mayor cantidad de errores.

Para finalizar, si tuviéramos que escoger un algoritmo y una hipótesis más la representación, con el análisis hecho se tomaría SMO usando las consulta médica y la representación TF-RFL.

13. Referencias

[1] Constantino Malagon Luque. Clasificadores bayesianos. El Algoritmo NaiveBayes.

[2] Wikipedia Precision and Recall Disponible: http://en.wikipedia.org/wiki/Precision_and_recall#Recall. Última visita: 10 de Septiembre de 2014.

[3]Diego A. Ingaramo, Marcelo L. Errecalde y Paolo Rosso. Medidas internas y externas en el agrupamiento de resúmenes científicos de dominios reducidos.

[4] ConfusionMatrix Disponible: http://www2.cs.uregina.ca/~dbd/cs831/notes/confusion_matrix/confusion_matrix.html.Última visita 11 de Septiembre de 2014.

[5] Andrew Ng, Support Vector Machines CS229 Lecture notes.

[6]John C. Platt. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines.

[7]GrigoriosTsoumakas, IoannisKatakis, and IoannisVlahavas.MiningMulti-label Data.

[8] Clasificación Estadística Internacional de Enfermedades y Problemas Relacionados con la Salud. Décima Revisión. Publicación Científica No. 554.

[9] Rodrigo Alfaro y Hector Allende. Text Representation in Multi-label Classification: Two New Input Representations.

[10] Jesse Read, Bernhard Pfahringer, Geoff Holmes, EibeFrank.Classifier Chains for Multi-label Classification.

[11]Hanna Suominen.Machine Learning and Clinical Text Supporting Health Information Flow Diciembre2009.

[12] S. B. KotsiantisDepartment of Computer Science and Technology University of Peloponnese, Greece End of Karaiskaki, 22100, TripolisGRSupervised Machine Learning: A Review of Classification Techniques.

[13] Daniel LowdyPedro Domingos.Naive Bayes Models for Probability Estimation.

[14]Thorsten Joachims. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization.

[15]Wikipedia K-Nearest Neighbors Algorithm.Disponible: http://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm. Últimavisita 15 de septiembre 2014.

[16] Wei-Yin Loh.Classification and regression trees.

[17]NelloCristianini, John Shawe Taylor. Introduction to Support Vector Machines.

[19]Classification Jesse Read, Bernhard Pfahringer,Geoff Holmes, Eibe Frank Department of Computer Science The University ofWaikato Hamilton, New Zealand.Classifier Chains for Multi-label

[20] Ian H. Witten, Eibe Frank, Mark A. Hall Data Mining: Practical Machine Learning Tools and Techniques, Third Edition.

14. Anexos

Anexo A: Tablas de Resultados Utilizados para Gráficos.

Tabla 14.1 : Códigos y Descripción Diagnóstico.

CÓDIGO CIE-10	DESCRIPCIÓN DIAGNÓSTICO
E11	Diabetes Mellitus
E78	Dislipidemia
I10	Hipertensión Esencial
Z00.1	Control de Salud
E66	Obesidad

Tabla 14.2 : Precisión Hipótesis 1.

PRECISIÓN			
	NAIVE	SVM	CC
TF-IDF E11	0,48	0,52	0,77
TF-RFL E11	0,66	0,56	0,90
TF-IDF E78	0,73	0,88	0,55
TF-RFL E78	0,77	0,97	0,96
TF-IDF I10	0,37	0,29	0,27
TF-RFL I10	0,37	0,44	0,14
TF-IDF Z00.1	0,72	0,85	0,55
TF-RFL Z00.1	0,77	0,94	0,84
TF-IDF E66	0,64	0,75	0,50
TF-RFL E66	0,69	0,74	0,52

Tabla 14.3: Recall Hipótesis 1

RECALL			
	NAIVE	SVM	CC
TF-IDF E11	0,57	0,79	0,43
TF-RFL E11	0,46	0,80	0,56
TF-IDF E78	0,63	0,75	0,21
TF-RFL E78	0,87	0,83	0,37
TF-IDF I10	0,39	0,14	0,14
TF-RFL I10	0,56	0,25	0,16
TF-IDF Z00.1	0,59	0,73	0,22
TF-RFL Z00.1	0,76	0,77	0,38
TF-IDF E66	0,42	0,50	0,15
TF-RFL E66	0,64	0,61	0,23

Tabla 14.4: Valor F Hipótesis 1.

VALOR F			
	NAIVE	SVM	CC
TF-IDF E11	0,52	0,63	0,56
TF-RFL E11	0,55	0,66	0,69
TF-IDF E78	0,68	0,81	0,30
TF-RFL E78	0,82	0,89	0,54
TF-IDF I10	0,38	0,19	0,18
TF-RFL I10	0,44	0,32	0,15
TF-IDF Z00.1	0,65	0,78	0,32
TF-RFL Z00.1	0,76	0,85	0,52
TF-IDF E66	0,57	0,60	0,23
TF-RFL E66	0,67	0,67	0,52

Tabla 14.5: Precisión Hipótesis 2.

PRECISIÓN			
	NAIVE	SVM	CC
TF-IDF E11	0,57	0,56	0,81
TF-RFL E11	0,71	0,68	0,99
TF-IDF E78	0,53	0,87	0,73
TF-RFL E78	0,77	0,95	0,96
TF-IDF I10	0,37	0,29	0,32
TF-RFL I10	0,49	0,48	0,12
TF-IDF Z00.1	0,95	0,90	0,44
TF-RFL Z00.1	0,86	0,95	0,94
TF-IDF E66	0,64	0,71	0,45
TF-RFL E66	0,75	0,78	0,61

Tabla 14.6 : Recall Hipótesis 2.

RECALL			
	NAIVE	SVM	CC
TF-IDF	0,48	0,69	0,46
TF-RFL	0,64	0,73	0,61
TF-IDF	0,89	0,84	0,26
TF-RFL	0,93	0,91	0,43
TF-IDF	0,38	0,21	0,17
TF-RFL	0,55	0,49	0,23
TF-IDF	0,59	0,81	0,21
TF-RFL	0,78	0,90	0,45
TF-IDF	0,57	0,60	0,22
TF-RFL	0,70	0,72	0,30

Tabla 14.7 : Valor F Hipótesis 2.

VALOR F			
	NAIVES	SVM	CC
TF-IDF E11	0,52	0,62	0,58
TF-RFL E11	0,67	0,71	0,76
TF-IDF E78	0,66	0,85	0,38
TF-RFL E78	0,84	0,93	0,59
TF-IDF I10	0,37	0,25	0,22
TF-RFL I10	0,52	0,49	0,15
TF-IDF Z00.1	0,73	0,86	0,28
TF-RFL Z00.1	0,82	0,92	0,59
TF-IDF E66	0,60	0,65	0,22
TF-RFL E66	0,73	0,75	0,40

Tabla 14.8 : Precisión Hipótesis 3.

PRECISION			
	NAIVE	SVM	CC
TF-IDF E11	0,61	0,58	0,77
TF-RFL E11	0,74	0,73	0,98
TF-IDF E78	0,75	0,90	0,54
TF-RFL E78	0,83	0,98	0,98
TF-IDF I10	0,39	0,27	0,25
TF-RFL I10	0,53	0,47	0,07
TF-IDF Z00.1	0,91	0,91	0,46
TF-RFL Z00.1	0,92	0,95	0,94
TF-IDF E66	0,70	0,74	0,33
TF-RFL E66	0,81	0,84	0,55

Tabla 14.9 : Recall Hipótesis 3.

RECALL			
	NAIVE	SVM	CC
TF-IDF E11	0,54	0,65	0,41
TF-RFL E11	0,73	0,74	0,59
TF-IDF E78	0,89	0,88	0,19
TF-RFL E78	0,96	0,93	0,41
TF-IDF I10	0,48	0,24	0,11
TF-RFL I10	0,51	0,52	0,14
TF-IDF Z00.1	0,75	0,88	0,18
TF-RFL Z00.1	0,87	0,92	0,43
TF-IDF E66	0,68	0,63	0,10
TF-RFL E66	0,75	0,75	0,27

Tabla 14.10 : Valor F Hipótesis 3.

VALOR F			
	NAIVE	SVM	CC
TF-IDF E11	0,53	0,61	0,53
TF-RFL E11	0,74	0,73	0,74
TF-IDF E78	0,82	0,89	0,28
TF-RFL E78	0,89	0,96	0,58
TF-IDF I10	0,43	0,26	0,16
TF-RFL I10	0,52	0,49	0,09
TF-IDF Z00.1	0,82	0,90	0,26
TF-RFL Z00.1	0,89	0,93	0,59
TF-IDF E66	0,69	0,68	0,15
TF-RFL E66	0,78	0,79	0,36

Anexo B: Gráficos por Categoría Hipótesis 1.

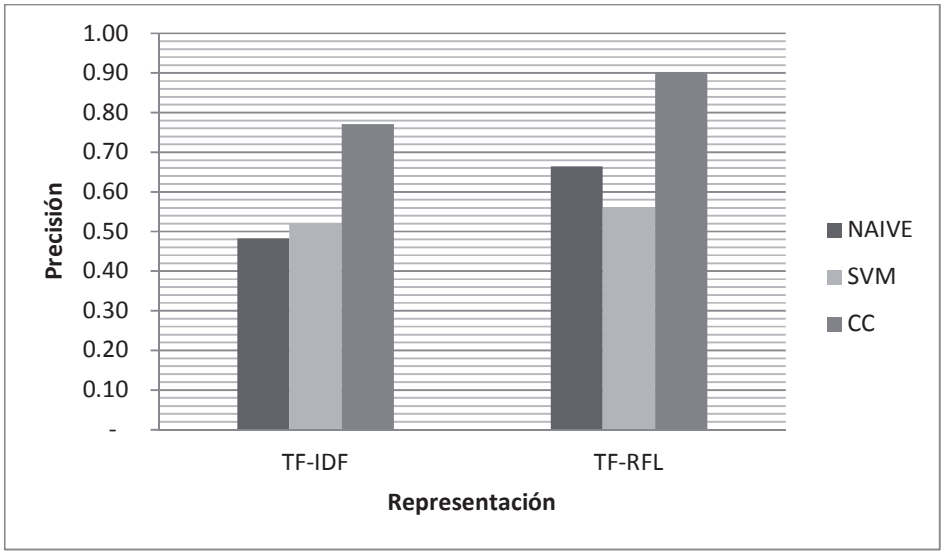

Figura 14.1 : Precisión en la categoría Diabetes Mellitus para la Hipótesis 1.

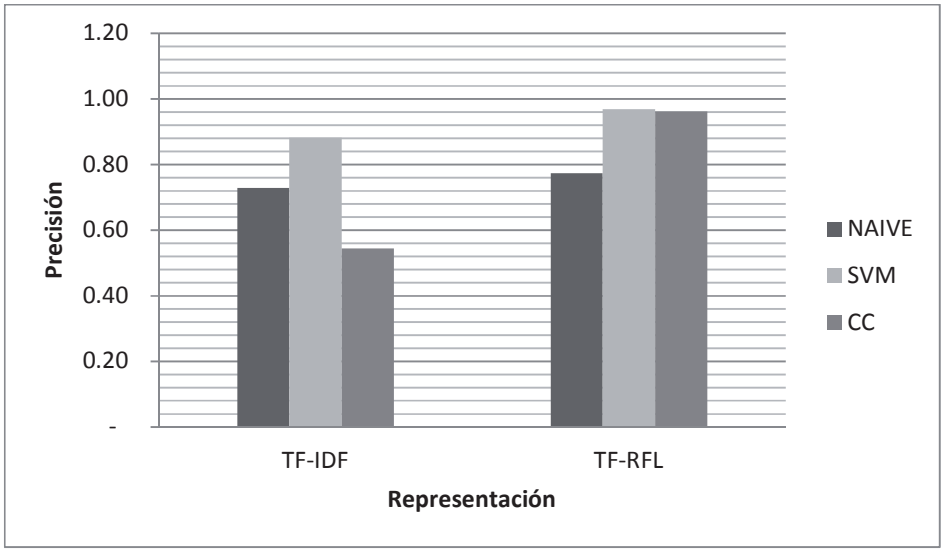

Figura 14.2 :Precisión en la categoría Dislipidemia para la Hipótesis 1.

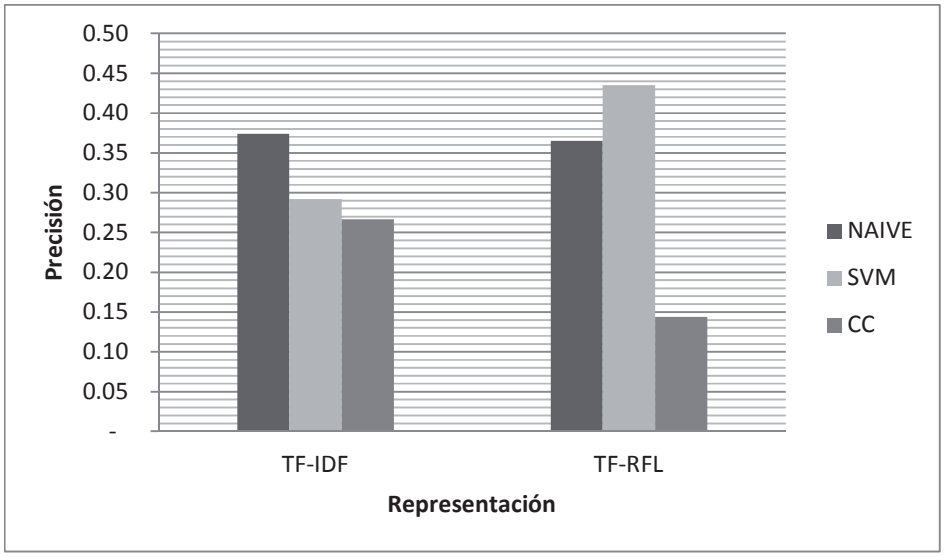

Figura 14.3 :Precisión en la categoría Hipertensión Esencial para la Hipótesis 1.

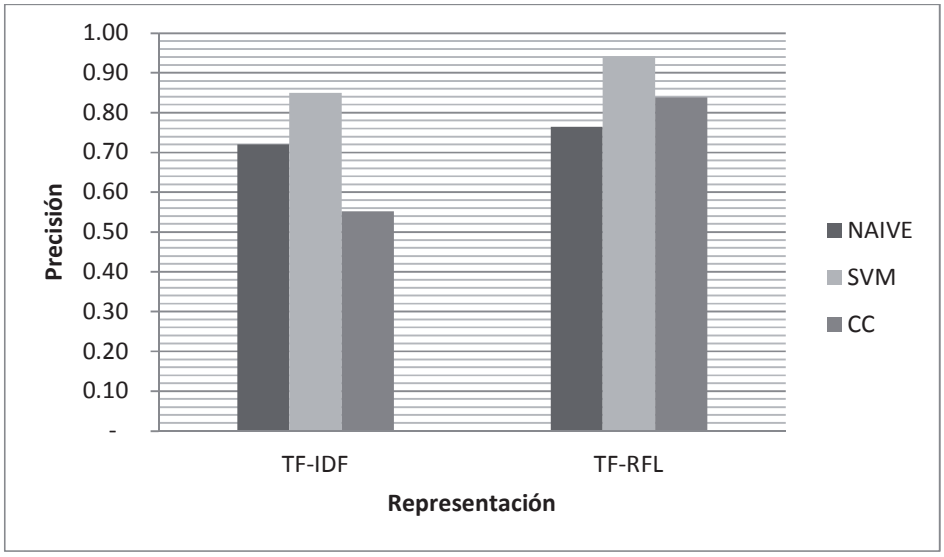

Figura 14.4 :Precisión en la categoría Control de Salud para la Hipótesis 1.

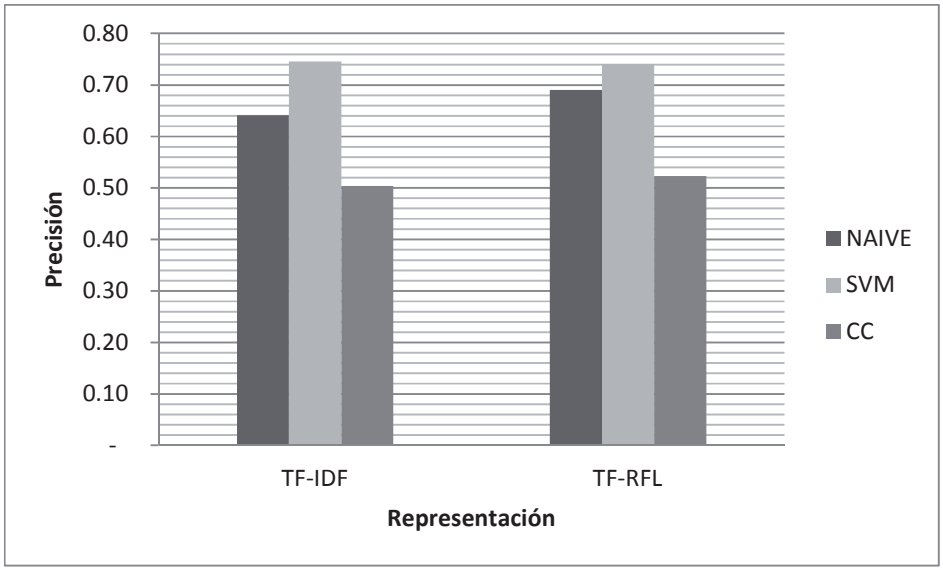

Figura 14.5 : Precisión en la categoría Obesidad para la Hipótesis 1.

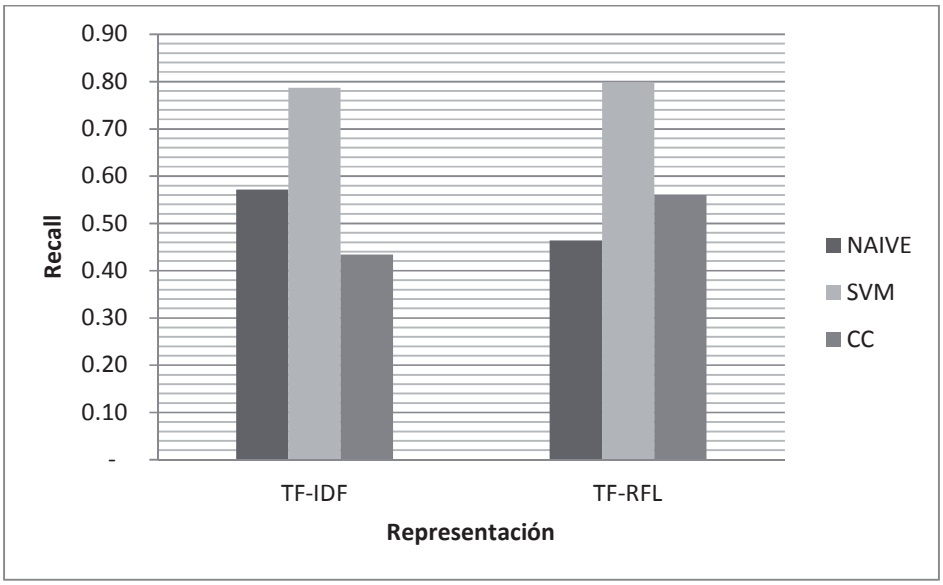

Figura 14.6 : Sensibilidad en la categoría Diabetes Mellitus para la Hipótesis 1.

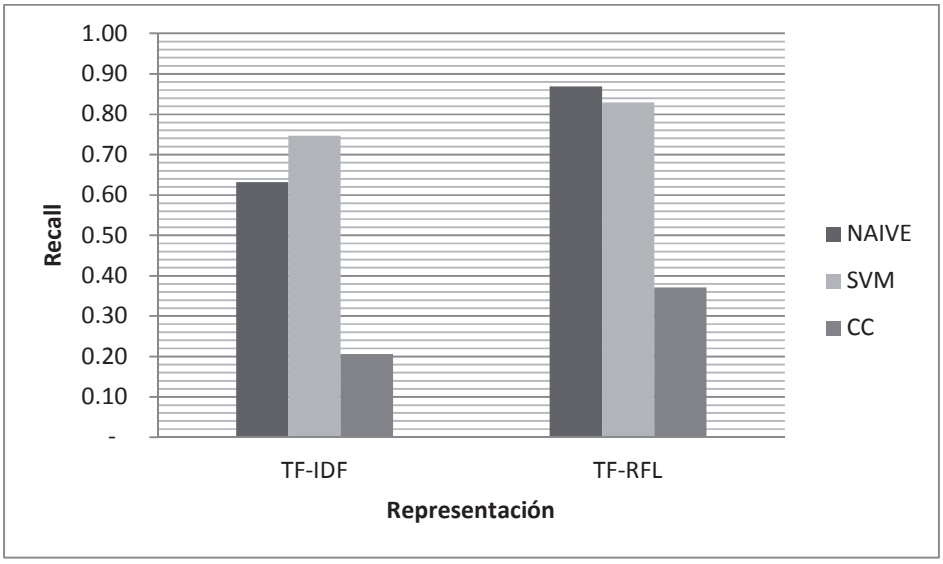

Figura 14.7 : Sensibilidad en la categoría Dislipidemia para la Hipótesis 1.

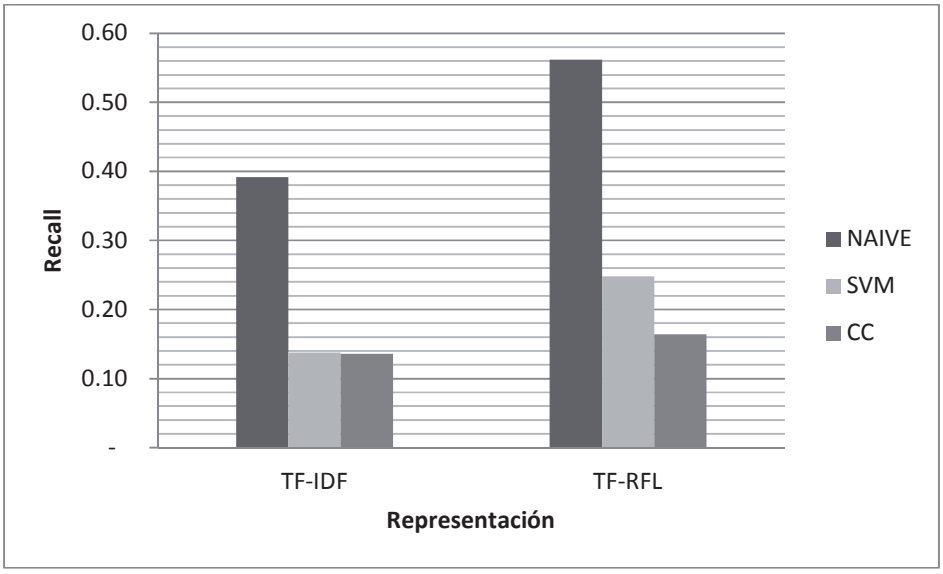

Figura 14.8 : Sensibilidad en la categoría Hipertensión Esencial para la Hipótesis 1.

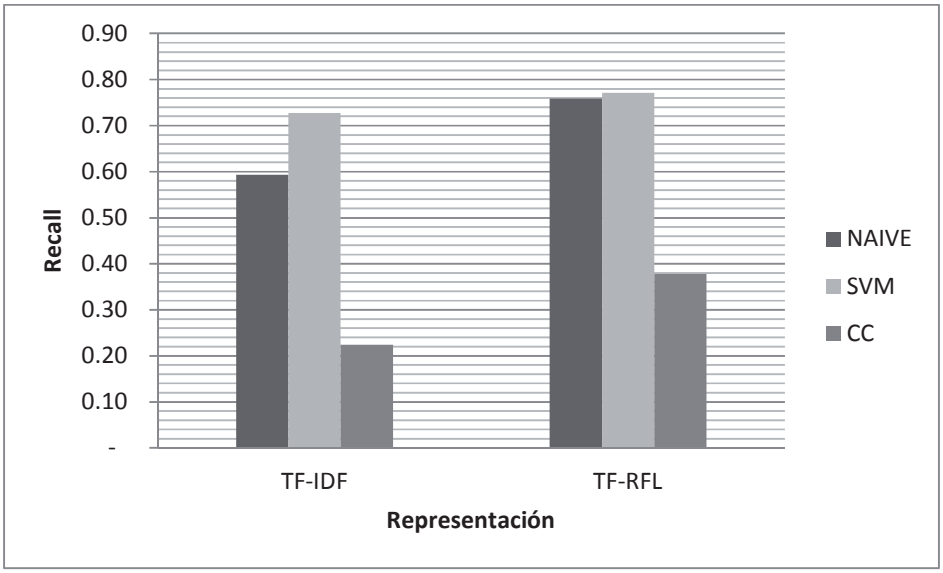

Figura 14.9 : Sensibilidad en la categoría Control de Salud para la Hipótesis 1.

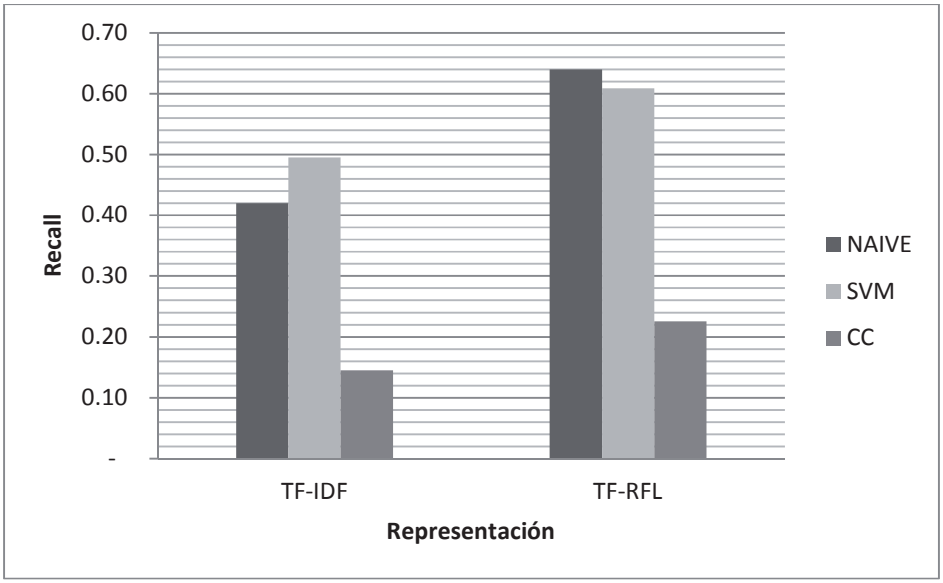

Figura 14.10 : Sensibilidad en la categoría Obesidad para la Hipótesis 1.

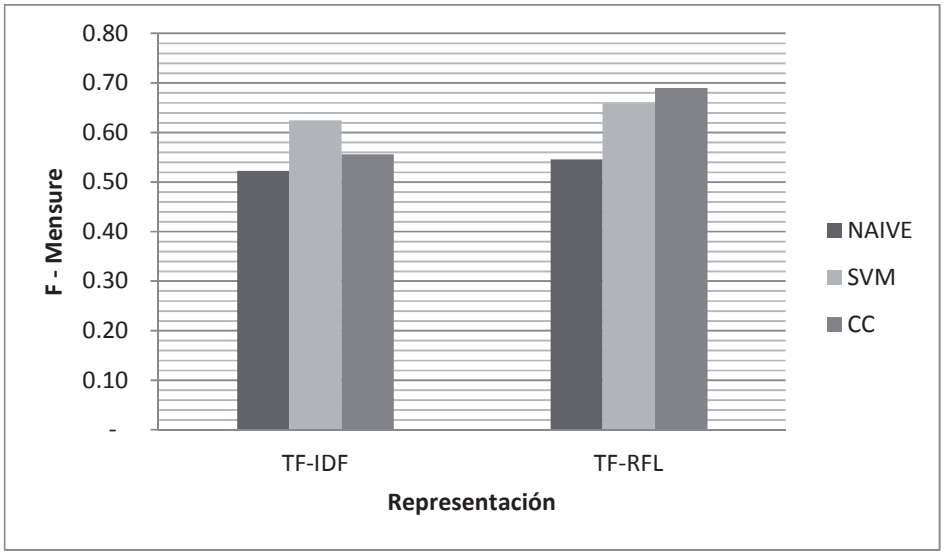

Figura 14.11 : Valor F en la categoría Diabetes Mellitus para la Hipótesis 1.

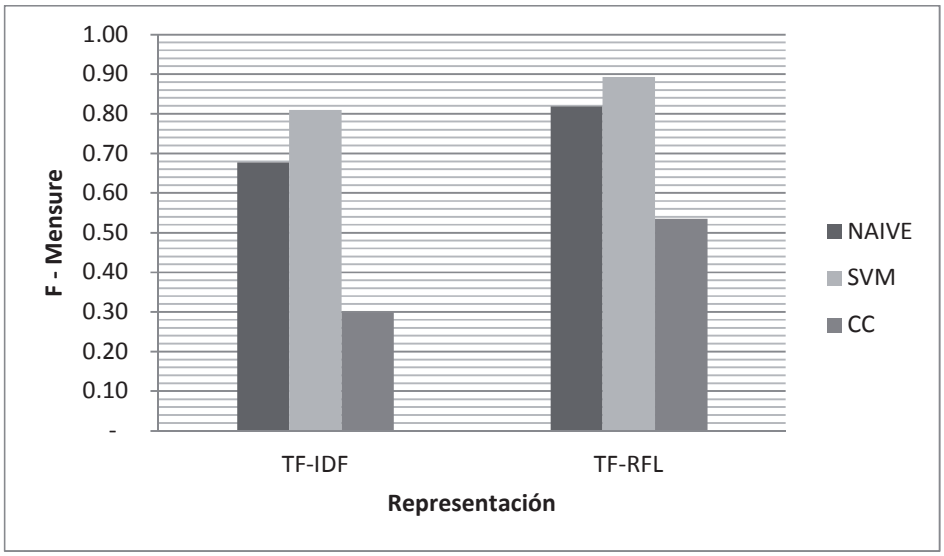

Figura 14.12 : Valor F en la categoría Dislipidemia para la Hipótesis 1.

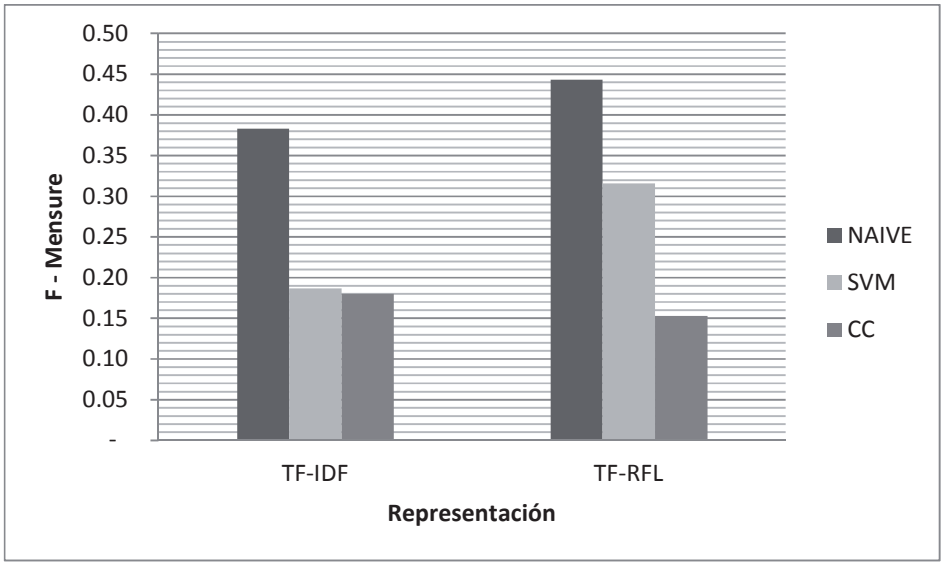

Figura 14.13 : Valor F en la categoría Hipertensión Esencial para la Hipótesis 1.

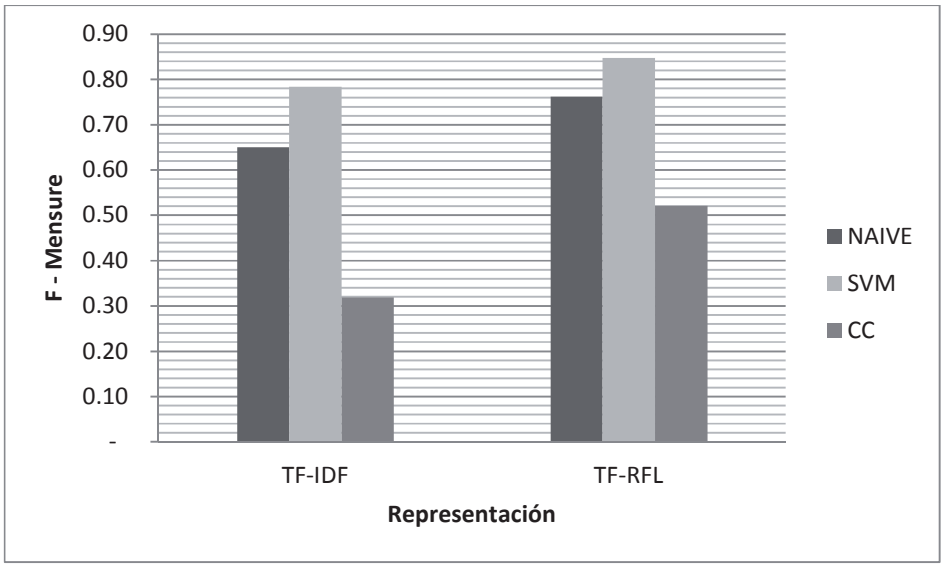

Figura 14.14 : Valor F en la categoría Control de Salud para la Hipótesis 1.

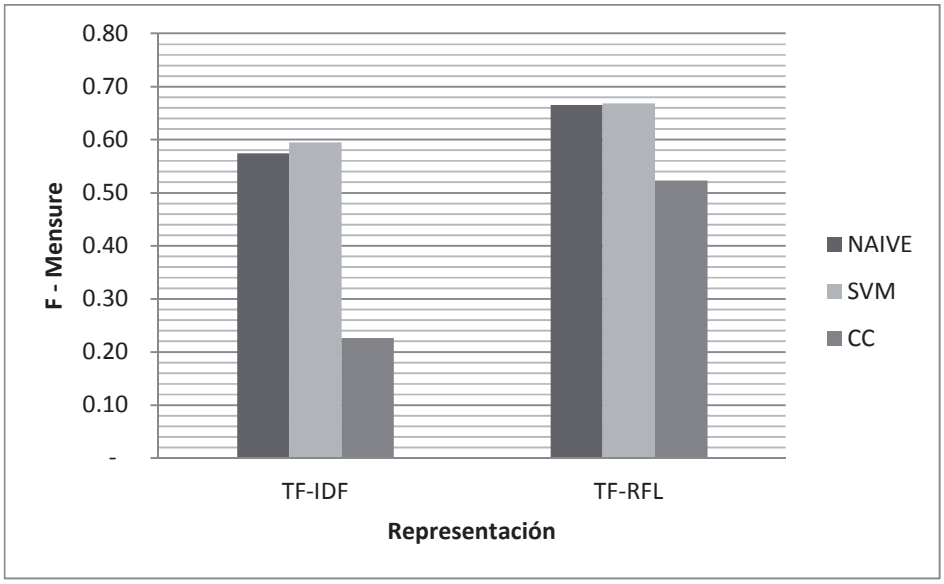

Figura 14.15 : Valor F en la categoría Obesidad para la Hipótesis 1.

Anexo C: Gráficos por Categoría Hipótesis 2.

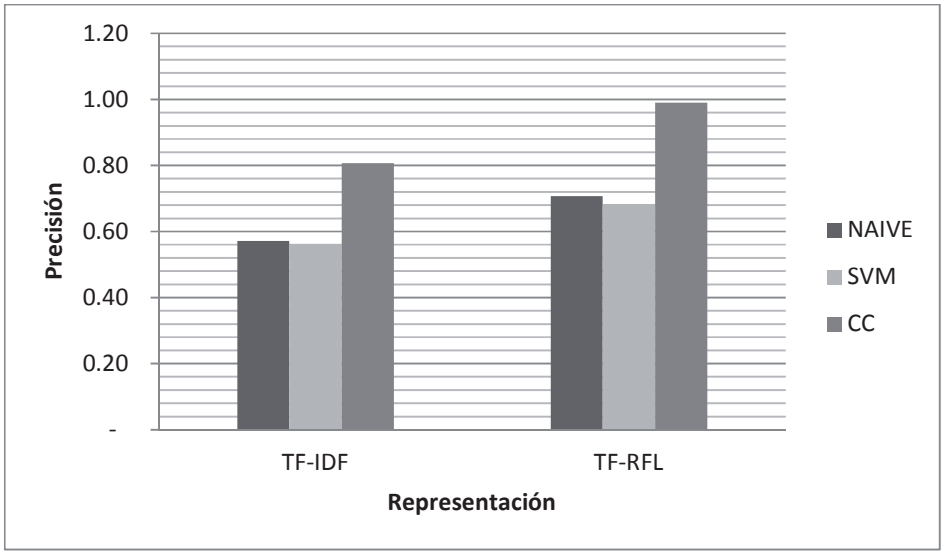

Figura 14.16 : Precisión en la categoría Diabetes Mellitus para la Hipótesis 2.

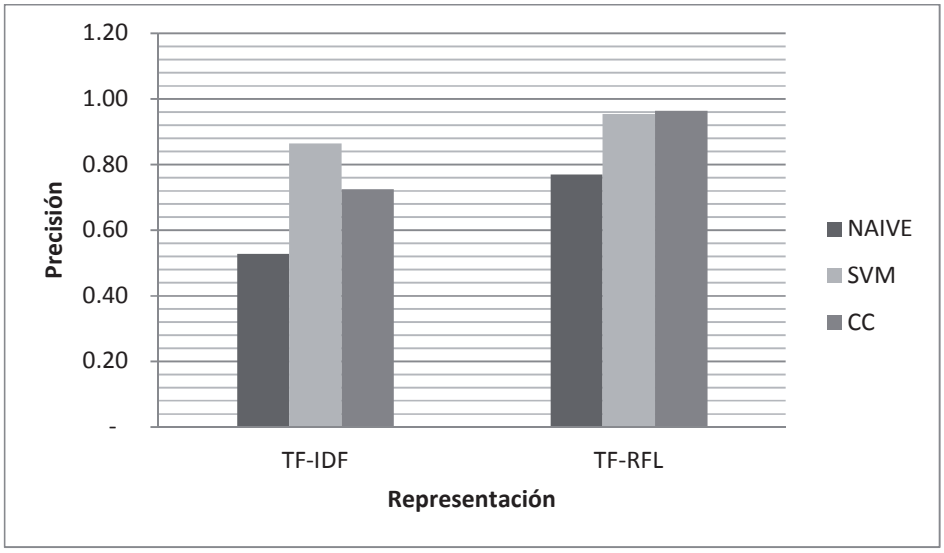

Figura 14.17 :Precisión en la categoría Dislipidemia para la Hipótesis 2.

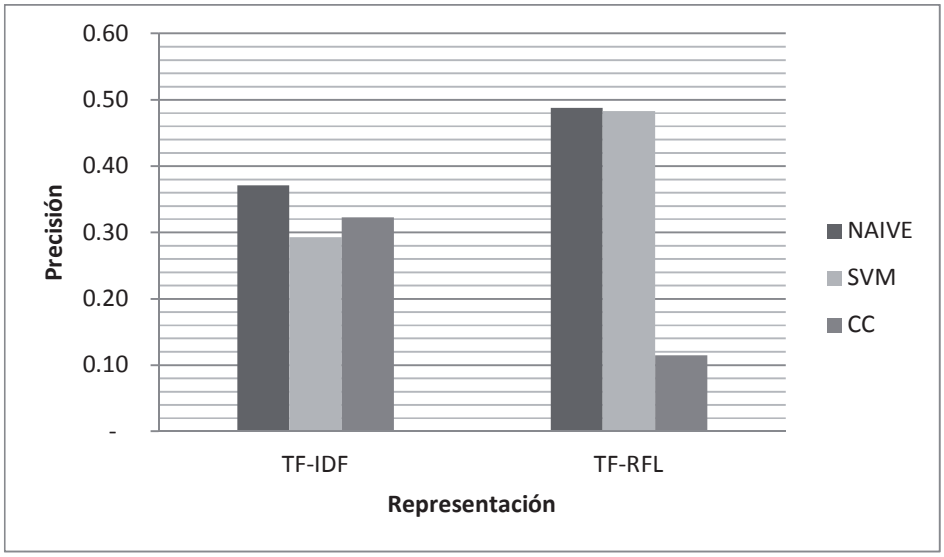

Figura 14.18 :Precisión en la categoría Hipertensión Esencial para la Hipótesis 2.

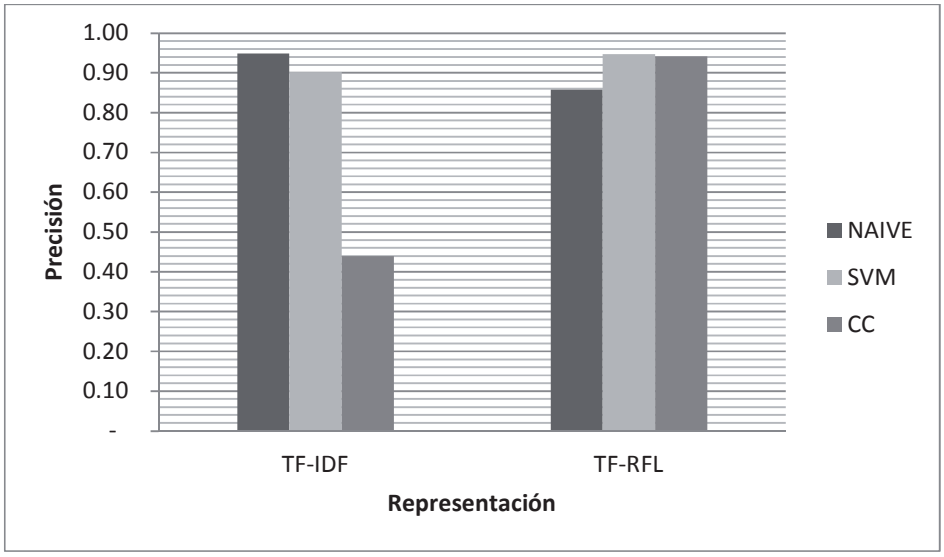

Figura 14.19 :Precisión en la categoría Control de Salud para la Hipótesis 2.

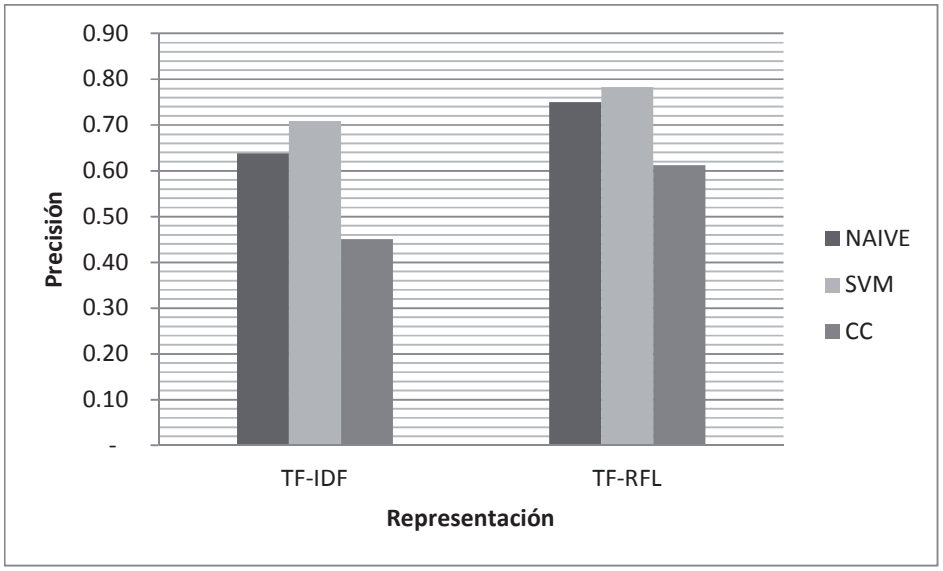

Figura 14.20 : Precisión en la categoría Obesidad para la Hipótesis 2.

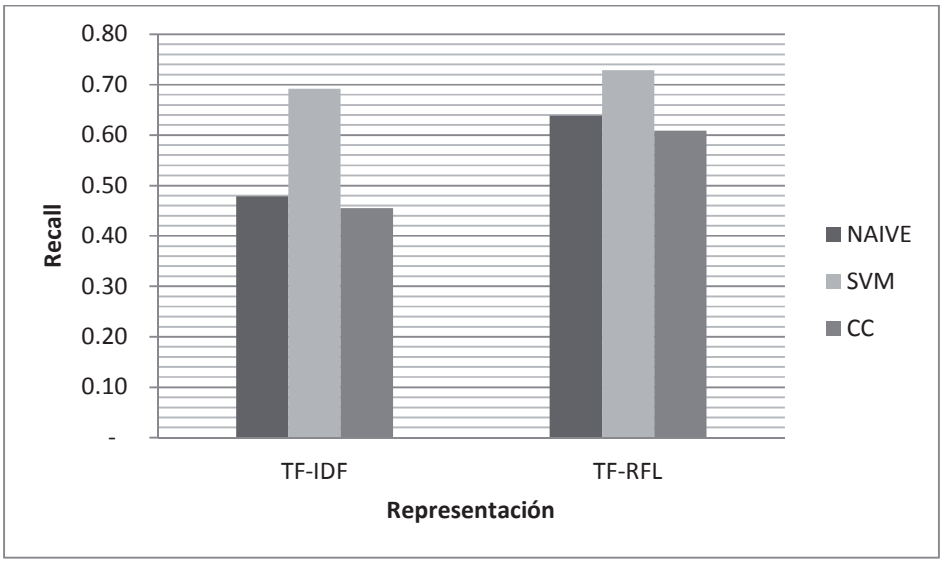

Figura 14.21 : Sensibilidad en la categoría Diabetes Mellitus para la Hipótesis 2.

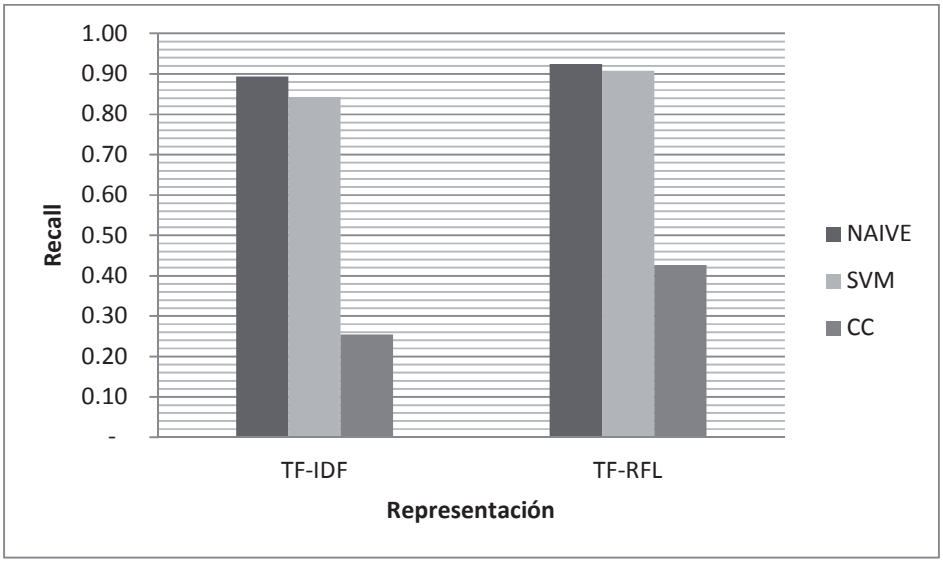

Figura 14.22 : Sensibilidad en la categoría Dislipidemia para la Hipótesis 2.

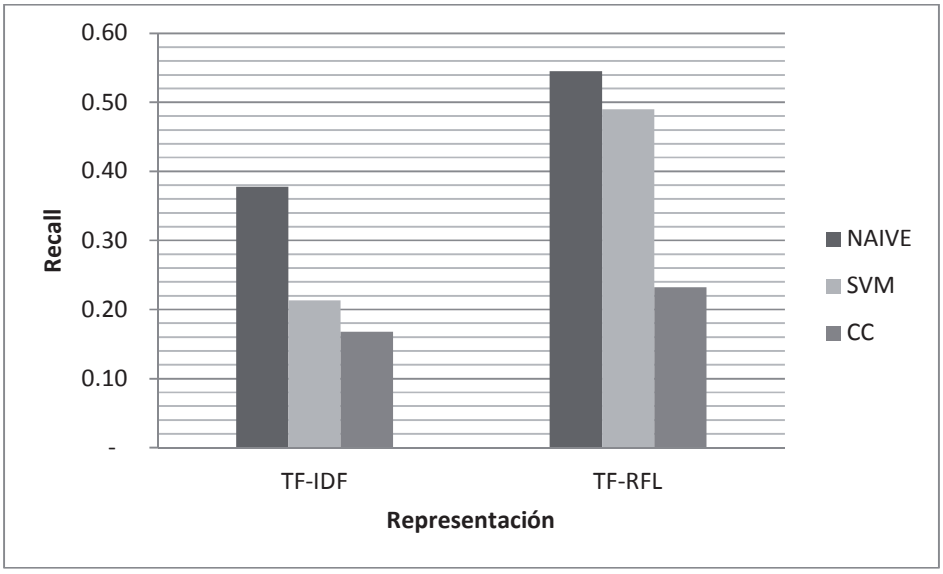

Figura 14.23 : Sensibilidad en la categoría Hipertensión Esencial para la Hipótesis 2.

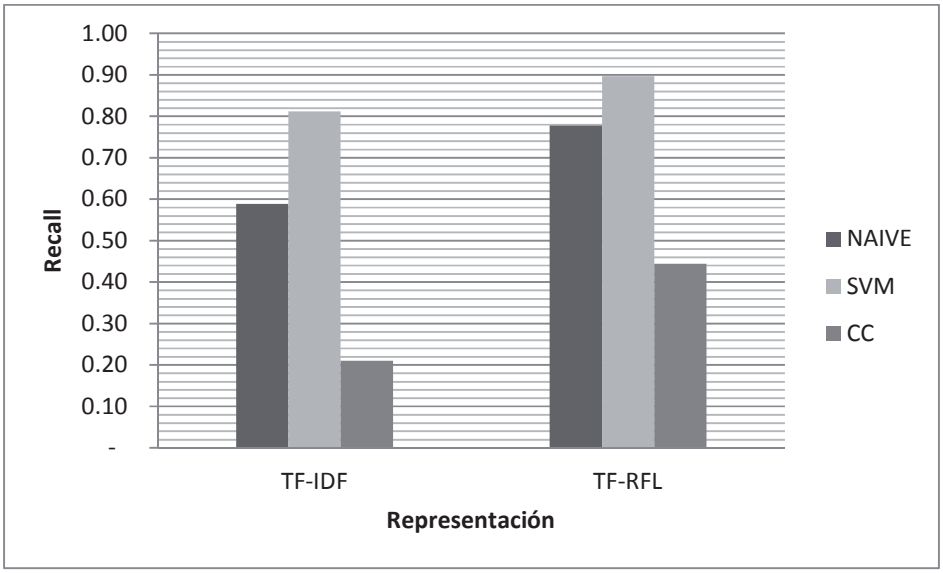

Figura 14.24 : Sensibilidad en la categoría Control de Salud para la Hipótesis 2.

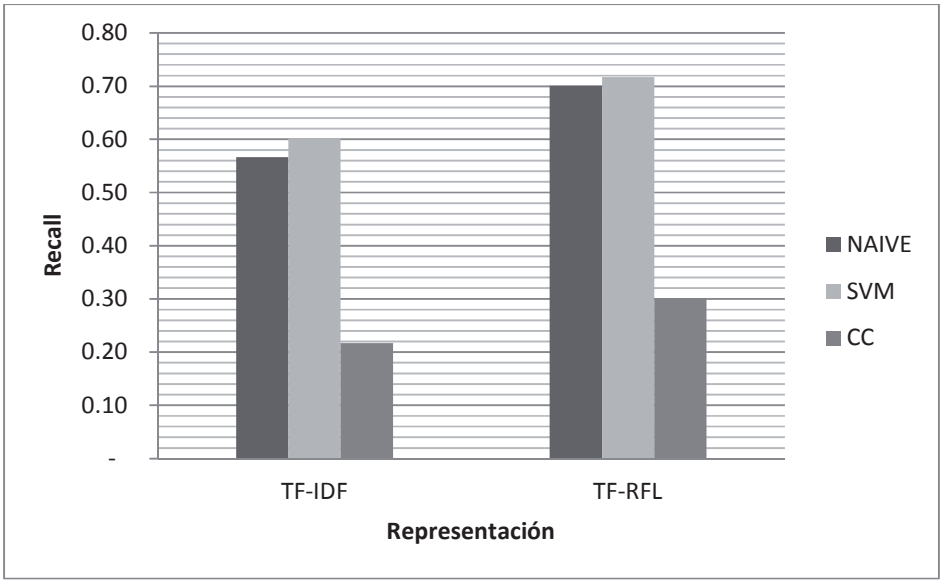

Figura 14.25 : Sensibilidad en la categoría Obesidad para la Hipótesis 2.

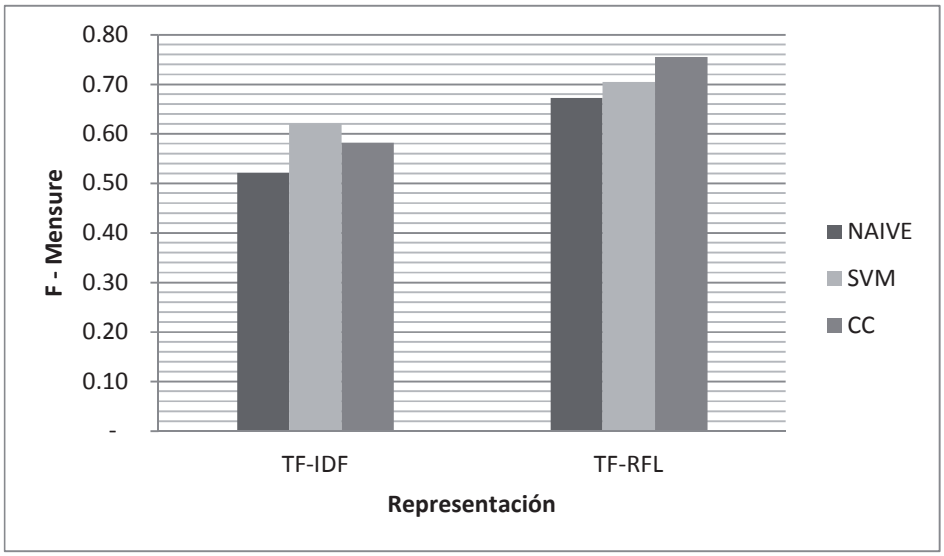

Figura 14.26 : Valor F en la categoría Diabetes Mellitus para la Hipótesis 2.

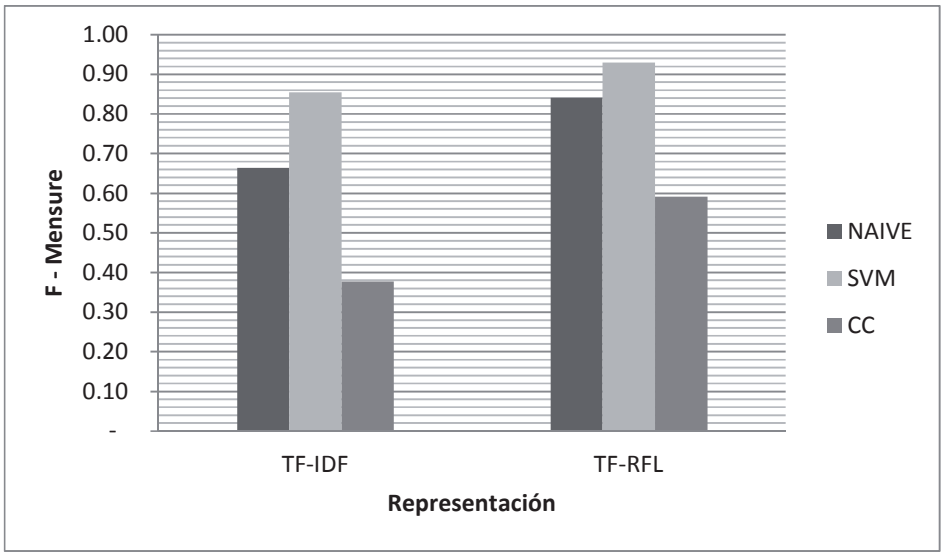

Figura 14.27 : Valor F en la categoría Dislipidemia para la Hipótesis 2.

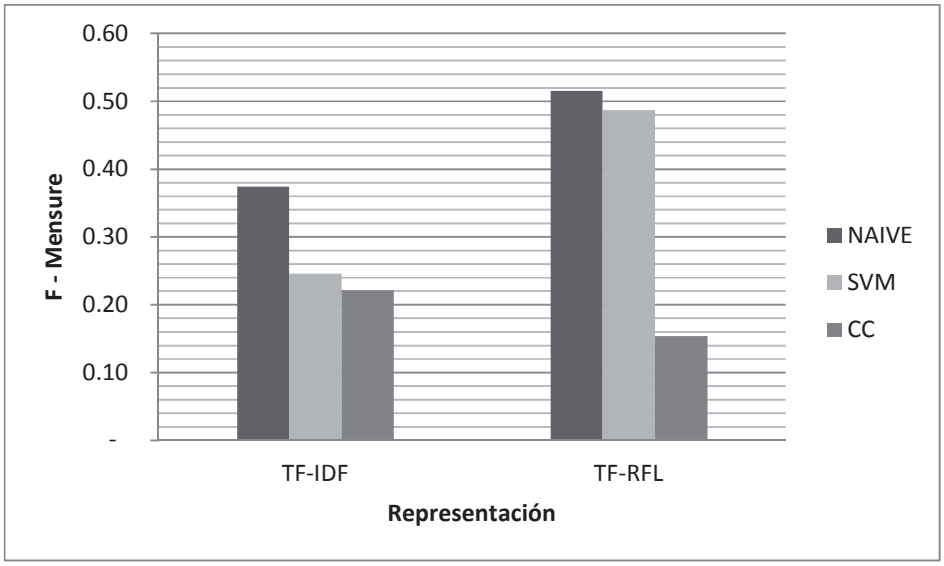

Figura 14.28 : Valor F en la categoría Hipertensión Esencial para la Hipótesis 2.

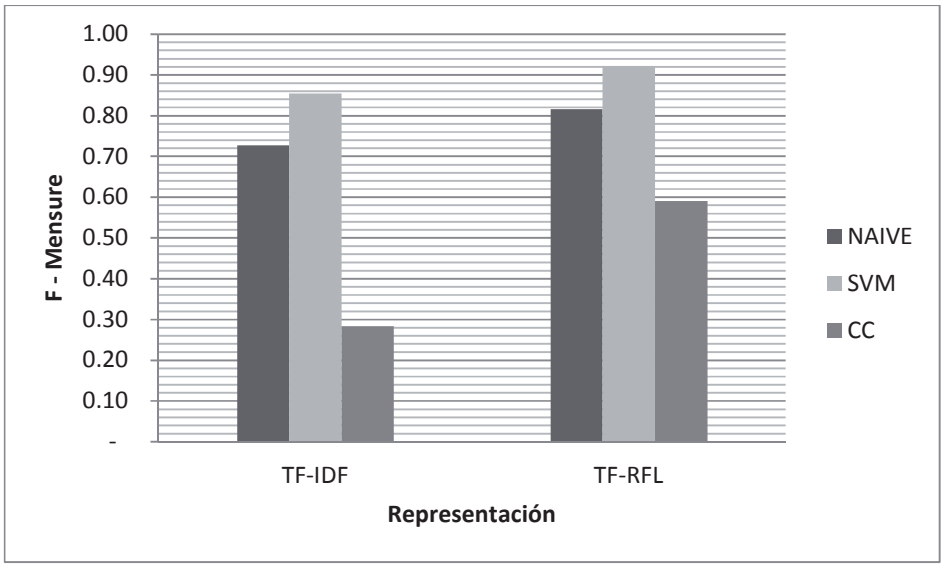

Figura 14.29 : Valor F en la categoría Control de Salud para la Hipótesis 2.

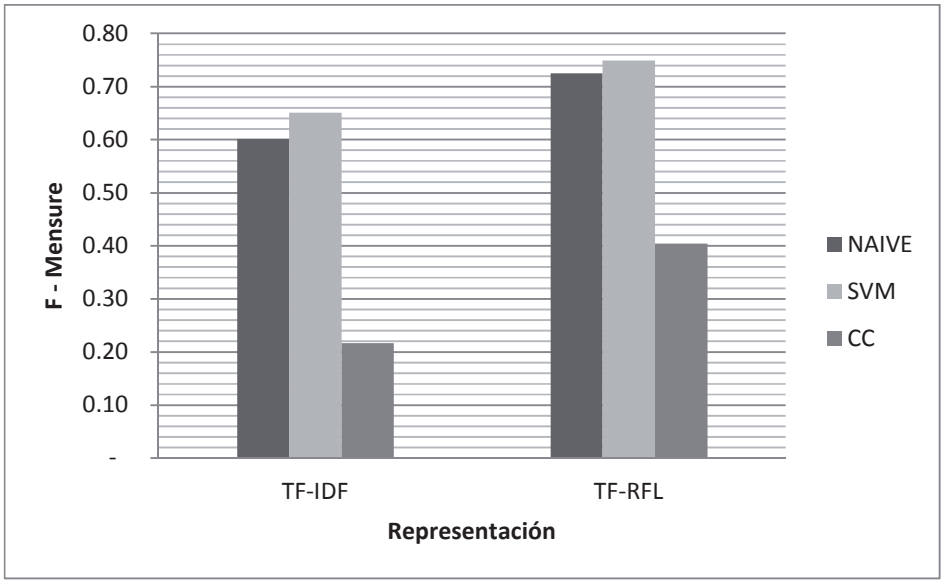

Figura 14.30 : Valor F en la categoría Obesidad para la Hipótesis 2.

Anexo D: Gráficos por Categoría Hipótesis 3.

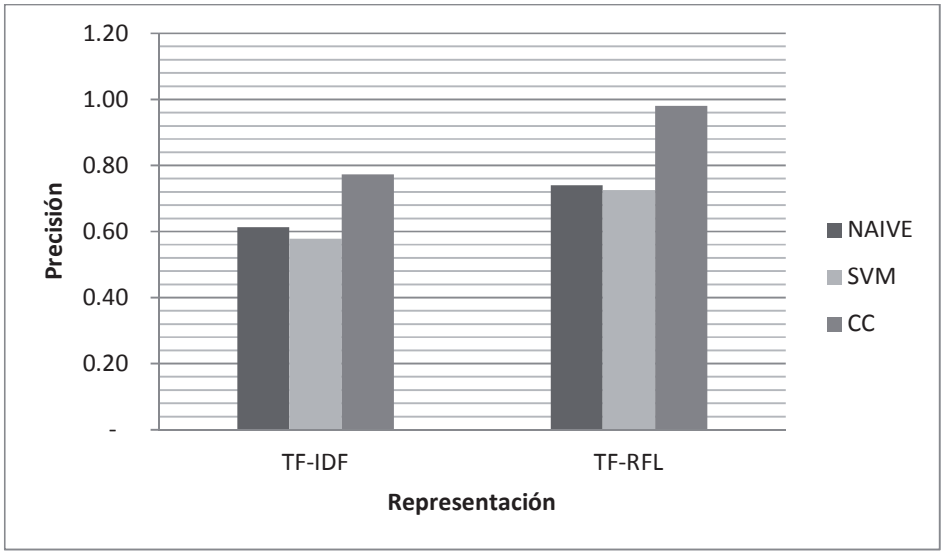

Figura 14.31 : Precisión en la categoría Diabetes Mellitus para la Hipótesis 3.

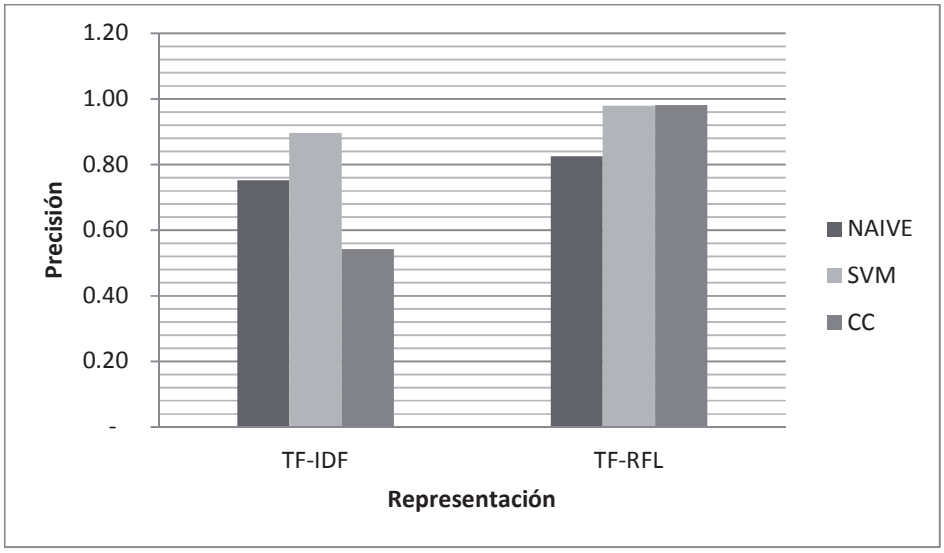

Figura 14.32 :Precisión en la categoría Dislipidemia para la Hipótesis 3.

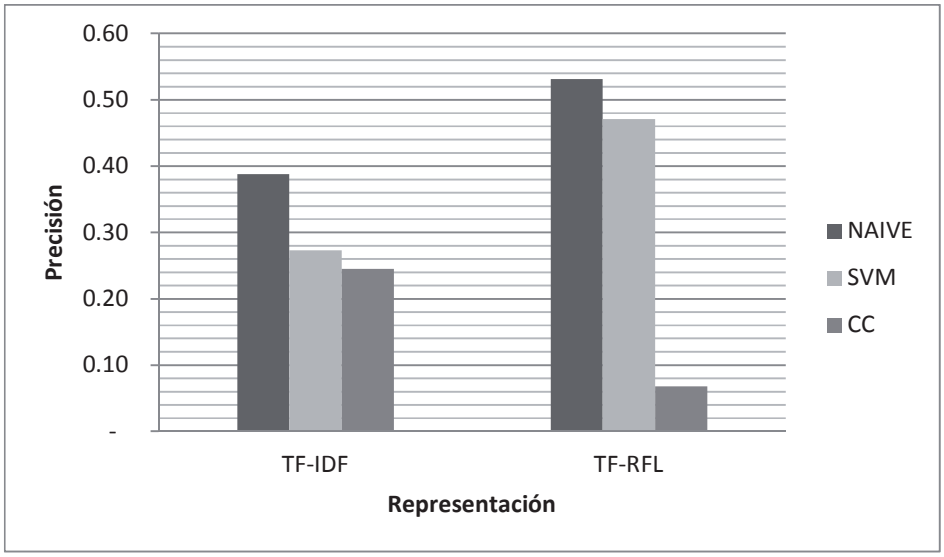

Figura 14.33 :Precisión en la categoría Hipertensión Esencial para la Hipótesis 3.

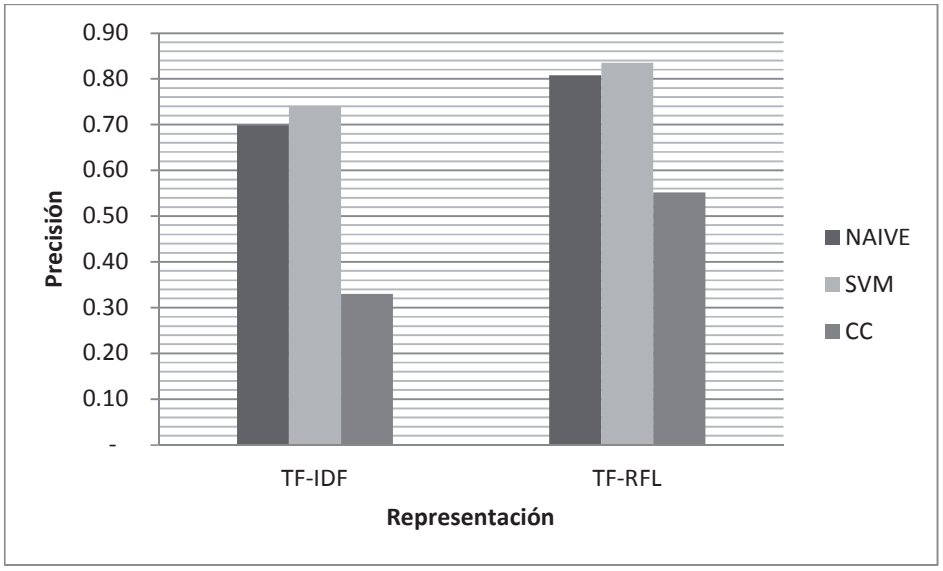

Figura 14.34 : Precisión en la categoría Obesidad para la Hipótesis 3.

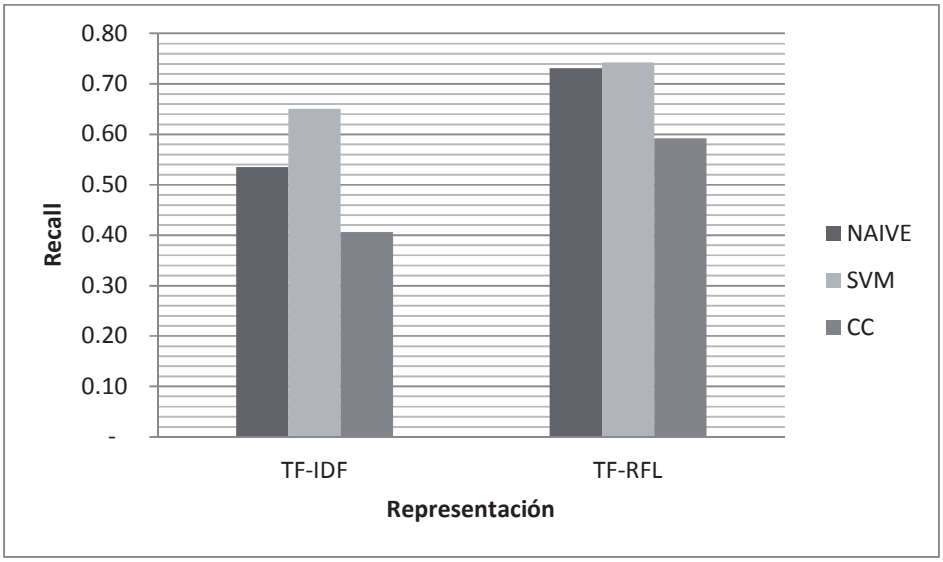

Figura 14.35 : Sensibilidad en la categoría Diabetes Mellitus para la Hipótesis 3.

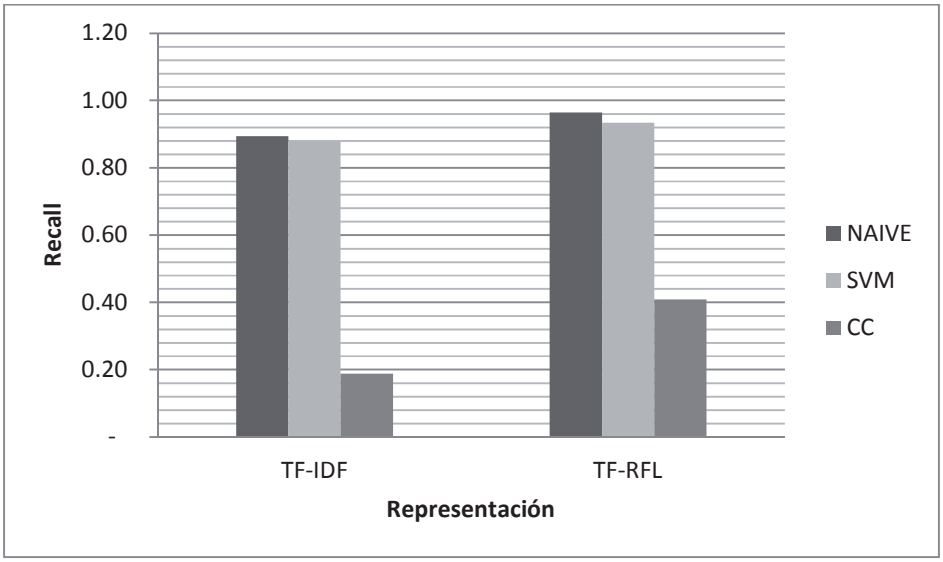

Figura 14.36 : Sensibilidad en la categoría Dislipidemia para la Hipótesis 3.

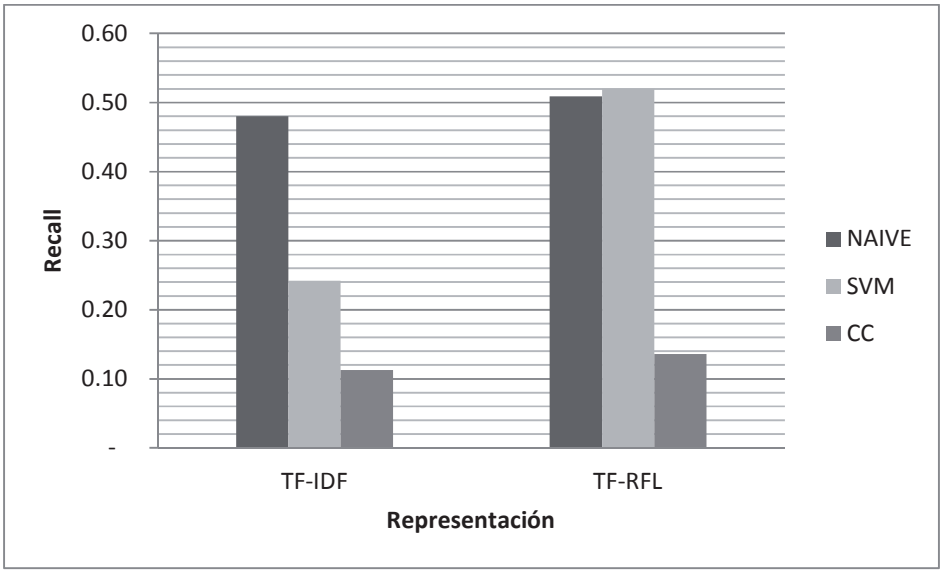

Figura 14.37 : Sensibilidad en la categoría Hipertensión Esencial para la Hipótesis 3.

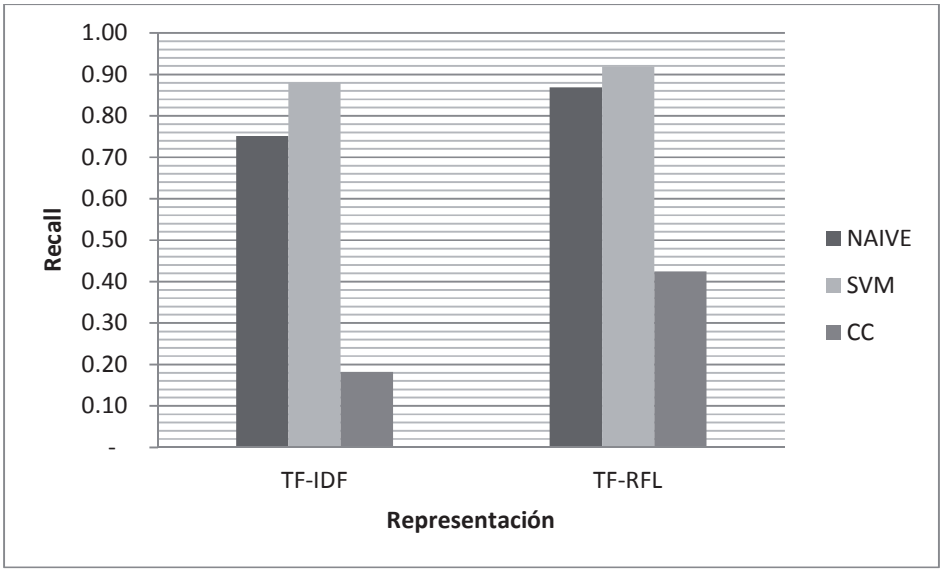

Figura 14.38 : Sensibilidad en la categoría Control de Salud para la Hipótesis 3.

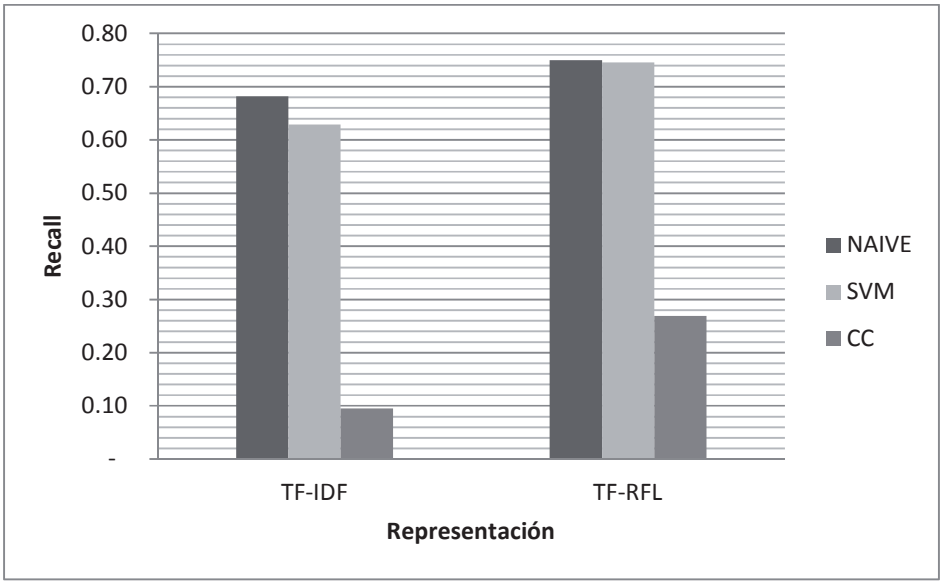

Figura 14.39 : Sensibilidad en la categoría Obesidad para la Hipótesis 3.

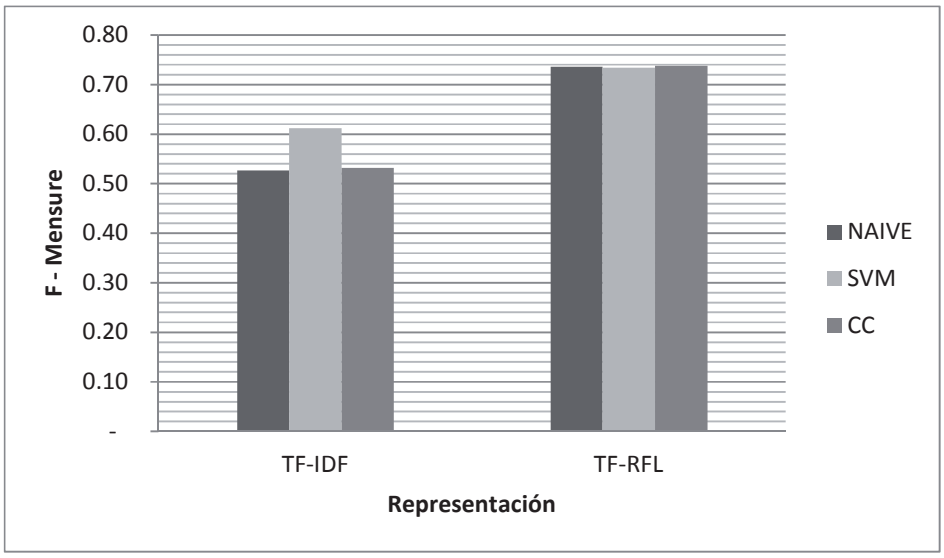

Figura 14.40 : Valor F en la categoría Diabetes Mellitus para la Hipótesis 3.

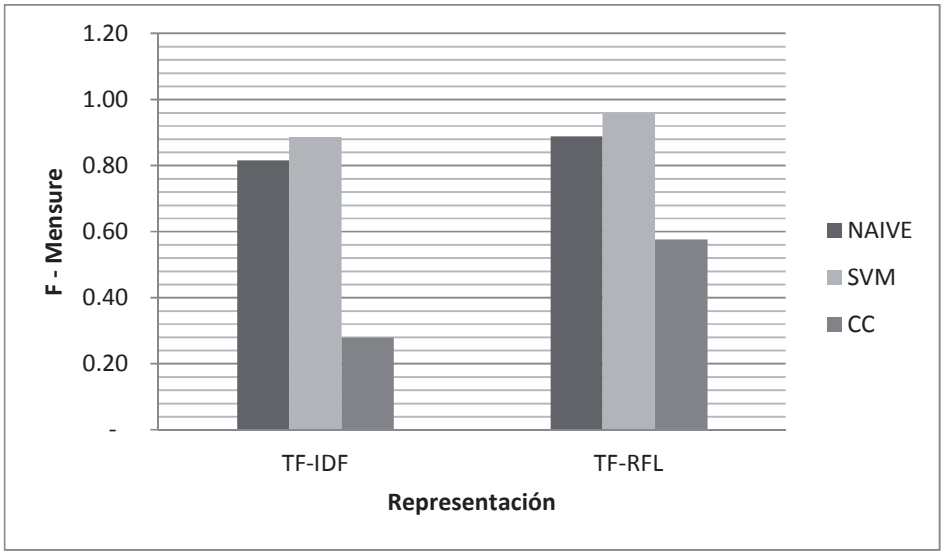

Figura 14.41 : Valor F en la categoría Dislipidemia para la Hipótesis 3.

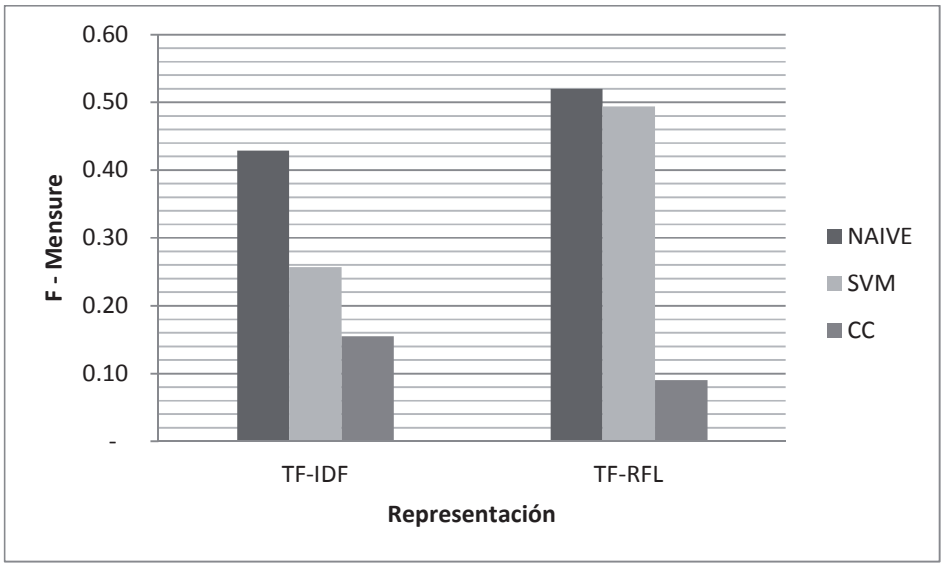

Figura 14.42 : Valor F en la categoría Hipertensión Esencial para la Hipótesis 3.

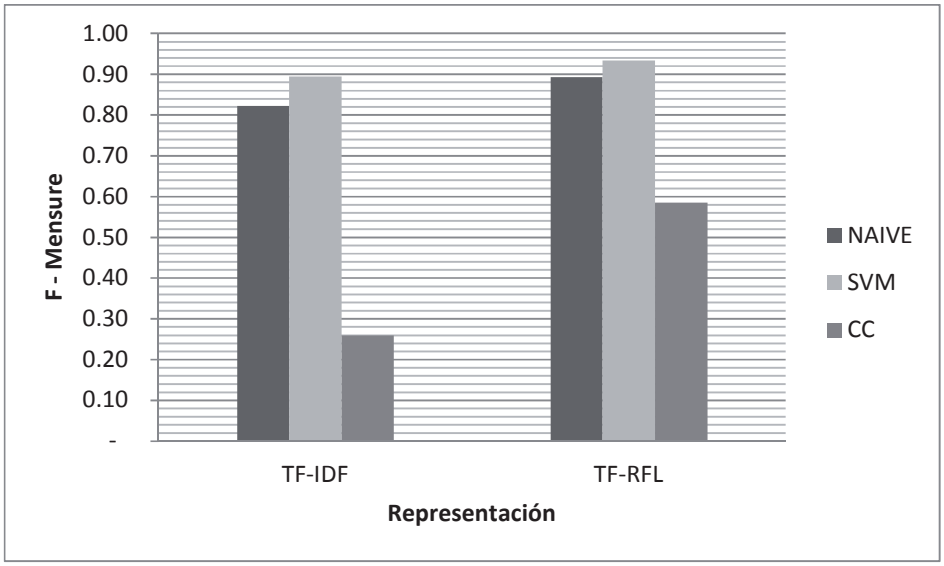

Figura 14.43 : Valor F en la categoría Control de Salud para la Hipótesis 3.

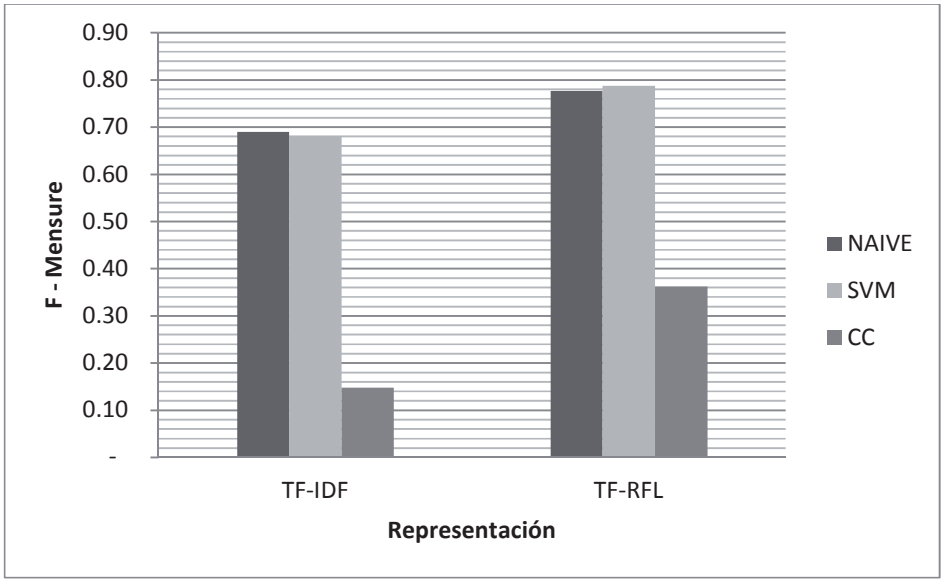

Figura 14.44 : Valor F en la categoría Obesidad para la Hipótesis 3.

Anexo E: Matriz de Resultados Hipótesis 1.

Tabla 14.11: Matriz de resultados hipótesis 1 algoritmo Cadena TF-IDF

	I10	E78	E11	Z00.1	E66
I10	1022	17	51	133	101
E78	116	349	53	75	47
E11	375	7	180	57	54
Z00.1	154	25	55	383	76
E66	119	9	32	74	238

Tabla 14.12: Matriz de resultados hipótesis 1 algoritmo Cadena TF-RFL

	I10	E78	E11	Z00.1	E66
I10	1193	101	7	18	5
E78	24	616	0	0	0
E11	540	33	97	1	2
Z00.1	30	59	0	581	23
E66	183	23	3	16	247

Tabla 14.13: Matriz de resultados hipótesis 1 algoritmo Naive Bayes TF-IDF

	I10	E78	E11	Z00.1	E66
I10	756	91	349	49	80
E78	233	425	1	13	0
E11	298	21	253	32	42
Z00.1	144	23	1	285	28
E66	135	23	72	268	238

Tabla 14.14:Matriz de resultados hipótesis 1 algoritmo Naive Bayes TF-RFL

	I10	E78	E11	Z00.1	E66
I10	618	100	467	59	89
E78	29	570	47	10	0
E11	208	34	378	25	28
Z00.1	19	17	65	384	21
E66	57	15	78	24	309

Tabla 14.15: Matriz de resultados hipótesis 1 algoritmo SMO TF-IDF

	I10	E78	E11	Z00.1	E66
I10	1043	37	178	20	47
E78	149	502	8	9	4
E11	518	11	89	7	21
Z00.1	103	10	3	350	15
E66	198	9	27	26	255

Tabla 14.16: Matriz de resultados hipótesis 1 algoritmo SMO TF-RFL

	I10	E78	E11	Z00.1	E66
I10	1064	14	191	5	59
E78	99	544	7	4	2
E11	481	2	167	1	22
Z00.1	91	0	5	390	20
E66	159	2	14	14	294

Anexo F: Matriz de Resultados Hipótesis 2.

Tabla 14.17:Matriz de resultados hipótesis 2 algoritmo Cadena TF-IDF

	I10	E78	E11	Z00.1	E66
I10	1069	144	24	30	57
E78	138	465	26	6	6
E11	310	81	218	34	30
Z00.1	254	23	21	305	90
E66	163	54	18	24	213

Tabla 14.18:Matriz de resultados hipótesis 2 algoritmo Cadena TF-RFL

	I10	E78	E11	Z00.1	E66
I10	1313	7	4	0	0
E78	23	618	0	0	0
E11	588	6	78	0	1
Z00.1	12	1	0	653	27
E66	162	5	3	13	289

Tabla 14.19:Matriz de resultados hipótesis 2 algoritmo Naive Bayes TF-IDF

	I10	E78	E11	Z00.1	E66
I10	675	320	375	1	37
E78	55	613	4	3	11
E11	282	105	247	2	18
Z00.1	78	40	2	314	99
E66	91	82	38	11	291

Tabla 14.20:Matriz de resultados hipótesis 2 algoritmo Naive Bayes TF-RFL

	I10	E78	E11	Z00.1	E66
I10	892	147	286	28	43
E78	18	633	16	5	12
E11	262	18	372	9	22
Z00.1	25	15	34	406	42
E66	63	9	55	25	357

Tabla 14.21: Matriz de resultados hipótesis 2 algoritmo SMO TF-IDF

	I10	E78	E11	Z00.1	E66
I10	974	54	301	8	71
E78	87	578	11	5	5
E11	480	12	139	2	21
Z00.1	52	14	4	433	30
E66	143	10	19	32	309

Tabla 14.22: Matriz de resultados hipótesis 2 algoritmo SMO TF-RFL

	I10	E78	E11	Z00.1	E66
I10	1017	25	297	2	55
E78	42	621	13	0	8
E11	326	0	335	1	21
Z00.1	21	2	14	468	17
E66	83	3	35	23	365

Anexo G: Matriz de Resultados Hipótesis 3.

Tabla 14.23:Matriz de resultados hipótesis 3 algoritmo Cadena TF-IDF

	I10	E78	E11	Z00.1	E66
I10	1022	17	30	145	108
E78	211	348	35	30	16
E11	340	12	165	81	75
Z00.1	245	39	15	316	78
E66	211	19	20	66	156

Tabla 14.24:Matriz de resultados hipótesis 3 algoritmo Cadena TF-RFL

	I10	E78	E11	Z00.1	E66
I10	1296	24	1	0	1
E78	11	629	0	0	0
E11	613	13	46	0	1
Z00.1	11	16	0	649	17
E66	184	10	2	15	261

Tabla 14.25:Matriz de resultados hipótesis 3 algoritmo Naive Bayes TF-IDF

	I10	E78	E11	Z00.1	E66
I10	771	117	460	18	74
E78	66	596	4	0	1
E11	274	42	317	5	22
Z00.1	51	22	2	389	54
E66	96	16	35	17	351

Tabla 14.26:Matriz de resultados hipótesis 3 algoritmo NaiveBayes TF-RFL

	I10	E78	E11	Z00.1	E66
I10	1052	92	236	16	44
E78	15	643	7	0	2
E11	287	16	336	8	13
Z00.1	16	12	7	450	33
E66	50	16	47	16	386

Tabla 14.27: Matriz de resultados hipótesis 3 algoritmo SMO TF-IDF

	I10	E78	E11	Z00.1	E66
I10	937	42	386	7	68
E78	67	586	7	3	4
E11	465	10	160	3	22
Z00.1	31	5	6	456	20
E66	121	11	27	32	324

Tabla 14.28: Matriz de resultados hipótesis 3algoritmo SMO TF-RFL

	I10	E78	E11	Z00.1	E66
I10	1070	9	311	5	45
E78	23	623	14	2	5
E11	304	0	343	1	12
Z00.1	8	1	19	476	14
E66	70	3	41	17	384

Anexo H: Rendimientos de Algoritmos.

Tabla 14.29: Rendimientos de los algoritmos.

ALGORITMO	REP.	HIPOTESIS	TIEMPO MODELO SEGUNDOS	TIEMPO TEST SEGUNDOS	N° PARAMETROS
SMO	IDF	HIS	1028	1	2093
SMO	IDF	CONSULTA	1023	1	2128
SMO	IDF	HIS+CONSULTA	1247	1	3529
SMO	RFL	HIS	478	1	2093
SMO	RFL	CONSULTA	568	1	2128
SMO	RFL	HIS+CONSULTA	647	1	3529
BAYES	IDF	HIS	14	16	2093
BAYES	IDF	CONSULTA	15	16	2128
BAYES	IDF	HIS+CONSULTA	7	25	3529
BAYES	RFL	HIS	5	15	2093
BAYES	RFL	CONSULTA	5	16	2128
BAYES	RFL	HIS+CONSULTA	7	28	3529
CADENA	IDF	HIS	12	18	13191
CADENA	IDF	CONSULTA	13	25	15011
CADENA	IDF	HIS+CONSULTA	35	89	24114
CADENA	RFL	HIS	12	18	13191
CADENA	RFL	CONSULTA	13	26	15011
CADENA	RFL	HIS+CONSULTA	36	90	24114