Universidad de las Ciencias Informáticas

Facultad 1

Clasificación supervisada de documentos mediante el algoritmo Naive Bayes

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Autor:

Alex Rodríguez Serrano

Tutores:

Ing. Yanedi Abreu Bartomeo

Ing. Luis Dominguez Cruz

La Habana, Junio del 2013

"El mundo camina hacia la era electrónica... Todo indica que esta ciencia se constituirá en algo así como una medida del desarrollo; quien la domine será un país de vanguardia. Vamos a volcar nuestros esfuerzos en este sentido con audacia revolucionaria."

Ernesto "Ché" Guevara

Por este medio declaro que soy el único autor de este trabajo y autorizo al Centro de Informatización Universitaria de la Universidad de las Ciencias Informáticas para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo el presente a los ___ días del mes de _____ del año _____.

_____ Firma del tutor
Ing.Luis Dominguez Cruz

_____ Firma del tutor
Ing.Yanedi Abreu Bartomeo

_____ Firma del autor
Alex Rodríguez Serrano

A mi madre, por su apoyo incondicional en todo lo que se me antoje y por confiar plenamente en mí.

A mi padre, por su total confianza, apoyo y ayuda a pesar de la distancia, sin esa ayuda hubiera sido imposible esta investigación.

A mi hermana por siempre estar ahí para mí.

A toda mi familia, los que están y los que no, por depositar siempre toda su confianza en mí y contribuir a mi formación como persona.

A mis amigos, tanto los de la UCI como los de Ciego, por confiar en mi amistad y brindarme su apoyo.

A Angel y Almarales por su apoyo en este trabajo.

A mis tutores por su ayuda.

A internet, por ser fuente indispensable de ayuda y consulta.

A todos aquellos que de una forma u otra han ayudado a mi formación profesional y personal y que no pongo por espacio.

El esfuerzo realizado durante mis años de estudio,
se lo dedico al sueño de mis padres y hermana,
de verme realizado como profesional.

Desde la antigüedad la información ha sido un elemento sumamente importante para el desarrollo de la humanidad. Por ello es de gran importancia almacenarla de una manera organizada para ahorrar tiempo y esfuerzos en el momento de consultarla.

En el repositorio institucional de la Universidad de las Ciencias Informáticas la organización de los documentos digitales no es la más correcta, por lo que se dificulta el proceso selección y adquisición de los mismos así como la navegabilidad del sistema de información del portal de la biblioteca.

El objetivo de esta investigación es implementar una clasificación supervisada de documentos, mediante el modelo gráfico probabilístico Naive Bayes en el repositorio documental de la Universidad de las Ciencias Informáticas, para contribuir a mejorar la navegabilidad.

Con el desarrollo de esta investigación se logra clasificar los documentos automáticamente en la clase a la cual pertenece cada uno de estos archivos. Una vez que el repositorio institucional cuente con este sistema de clasificación contara con una mejor organización de los documentos, lo que acompañado de un sistema de búsquedas temáticas facilitará que se puedan localizar los archivos por los temas que estos tratan, propiciando así que se mejore la navegabilidad del sistema de información en el portal de la biblioteca.

Palabras clave: Clasificación de documentos, documentos, Naive Bayes, repositorio.

Índice

INTRODUCCIÓN 1
Capítulo 1 Estudio sobre clasificación supervisada de documentos, en sistemas informáticos. 7
1.1 Introducción 7
1.2 Descripción de términos 7
1.3 Algoritmos de Clasificación Supervisada 10
1.4 Algoritmo Naive Bayes y su evolución 12
1.5 Estudios relacionados al algoritmo 14
1.6 Tecnologías necesarias para la implementación del algoritmo. 17
1.7 Conclusiones parciales 22
Capítulo 2: Análisis del clasificador de documentos basado en el modelo gráfico probabilístico Naive Bayes. 23
2.1 Introducción 23
2.2 Propuesta de solución 23
2.3 Selección de una técnica de suavizado 26
2.4 Pseudocódigos del algoritmo 27
2.5 Modelo de dominio 29
2.6 Requerimientos 30
2.7 Historias de Usuario 31
2.8 Estándares de codificación 36
2.9 Diagrama de clases 39
2.10 Conclusiones parciales 41
Capítulo 3 Implementación y pruebas del clasificador de documentos mediante el modelo Naive Bayes 42
3.1 Introducción 42
3.2 Descripción de las fases principales del clasificador 42
3.3 Análisis del código de la solución. 45
3.4 Pruebas 48
3.5 Conclusiones parciales 54
Conclusiones 55
Recomendaciones 56
Referencias Bibliográficas 57

Bibliografía consultada.....64
Glosario de términos.....68
Anexos.....69

ÍNDICE DE FIGURAS

Figura 1: Modelo de dominio. 29
Figura 2: Diagrama de clases. 40
Figura 3: Eliminar datos irrelevantes. 46
Figura 4: Calcular la probabilidad de que un documento pertenezca a una clase. 47
Figura 5: Calcular la probabilidad a priori de la clase. 48
Figura 6: Código del caso de prueba 1. Calcular probabilidad de pertenencia de un documento en una clase. 49
Figura 7: Grafo del caso de prueba 1. Calcular probabilidad de pertenencia de un documento en una clase. 49
Figura 8: Código del caso de prueba 2. Seleccionar máxima probabilidad. 50
Figura 9: Grafo del caso de prueba 2. Seleccionar máxima probabilidad. 50

ÍNDICE DE TABLAS

Tabla 1: Historia de Usuario clasificar documento. 32
Tabla 2: Historia de Usuario seleccionar atributos del documento. 33
Tabla 3: Historia de Usuario calcular probabilidades. 34
Tabla 4: Historia de Usuario asignar documento a la clase que pertenece. 35
Tabla 5: Grupos de la colección de documentos. 42
Tabla 6: Grupos seleccionados para la colección de entrenamiento. 43
Tabla 7: Resultados de la etapa de entrenamiento. 44
Tabla 8: Resultados de las pruebas de precisión. 1ra iteración. 52
Tabla 9: Resultados de las pruebas de precisión. 2da iteración. 53

INTRODUCCIÓN

Las Tecnologías de la Información y la Comunicación, también conocidas como TIC, son el conjunto de tecnologías desarrolladas para gestionar información y enviarla de un lugar a otro. Estas abarcan un abanico de soluciones muy amplio que incluyen las tecnologías para almacenar información y recuperarla después (ECHEVERRÍA *et al.*, 2012).

Estas Tecnologías conjuntamente con Internet han logrado un desarrollo acelerado en los primeros años del siglo actual. De acuerdo a las estadísticas de *Internet World Stats* (WIS) las regiones más beneficiadas en cuanto al crecimiento de Internet son Asia, Europa, América del Norte y América Latina y el Caribe (WIS, 2012).

Dentro de las principales tendencias que se evidencian con este aumento de Internet se encuentra la informatización de la sociedad. Esta inclinación es el proceso de utilización de las Tecnologías de la Información y la Comunicación en la vida cotidiana, para facilitar las tareas de crear, acceder, utilizar y compartir información; satisfaciendo las necesidades de todas las esferas de la población. El interés por la informatización proviene del esfuerzo de la sociedad por lograr cada vez más eficacia y eficiencia en todos los procesos y por consiguiente mayor generación de riqueza y aumento en la calidad de vida de los ciudadanos.

En el marco de la informatización de la sociedad se encuentra la creación de la biblioteca digital. Son variadas las reflexiones acerca del concepto de este término, dentro de estas resalta una realizada por el maestro Voutssás (2002) cuando señalaba que: "Biblioteca digital es una evolución de conceptos, de uso de términos emergentes en la literatura, de neologismos que aparecen sucesivamente... y, a la larga, todo eso se conjuga después de una serie de discusiones, los conceptos comienzan a aterrizarse y las bibliotecas digitales se hacen más reales; en consecuencia, se hacen herederas de esos términos que estaban ahí. La biblioteca digital depende de quién la describe, su proceso de construcción es interdisciplinario, no solo es una actividad para bibliotecarios, sino también para expertos en comunicaciones, ingenieros en sistemas, creadores de bases de datos, es decir, existe una participación de intereses de muchos sectores comerciales, académicos, técnicos, etc., y muchos enfoques". En este tipo de biblioteca,

las tareas de acceso, adquisición y almacenamiento de información se llevan a cabo a través de tecnologías digitales.

Cuba, que produce bienes y servicios dirigidos a la satisfacción de las necesidades crecientes de la población, desde la creación de los Joven Club de Computación en 1987 se encuentra inmersa en la campaña de informatización de la sociedad.

En el año 2000 con el propósito de reunir la obra escrita de autores cubanos entre los siglos XVII-XIX surge la Biblioteca Digital de Cuba, también conocida por biblioteca digital José Martí. Su creación evidencia que el país avanza con sus escasos recursos junto a varios países latinoamericanos en los aspectos tecnológicos.

La Universidad de las Ciencias Informáticas (UCI), a pesar de ser la universidad más joven del país, se ha adueñado de uno de los primeros lugares en cuanto a creación de software en Cuba. Por eso, la comunidad universitaria se vio en la necesidad de tener a su alcance una biblioteca digital que facilitara la consulta de documentos electrónicos, aunque hasta la actualidad y basándose en el concepto, en la universidad se cuenta con una “biblioteca híbrida”.

Una biblioteca híbrida es la reunión de tecnologías: electrónica, digital o virtual más los productos impresos y servicios en espacio físico y las funciones históricas de ésta (VILLA y ALFONSO, 2005).

La creación y actualización del portal de la Biblioteca de la Universidad de las Ciencias Informáticas, se encuentra dentro del radio de acción del Centro de Informatización Universitaria cuya misión es la informatización de la gestión universitaria y el desarrollo del concepto de universidad digital.

El portal cuenta con un amplio catálogo de recursos, por lo cual todos los miembros de la comunidad universitaria se benefician del repositorio institucional para realizar tareas, investigaciones o simplemente para aumentar su nivel intelectual. Para facilitar la utilización del sumario documental y atraer a los usuarios, se implementó un sistema de búsqueda de documentos.

En la actualidad el mencionado repositorio utiliza un sistema de clasificación de tipo conceptual. Este tipo de clasificación se basa en la categorización por las palabras clave contenidas en cada uno de los documentos que se introducen en el para beneficio de los usuarios que acceden a este sistema.

El uso de una catalogación conceptual en vez de una clasificación temática o por categoría siendo la segunda según Hassan y Martín (2004), “la más útil de las clasificaciones, en la que se trata de organizar los contenidos en categorías definidas en función de la temática de los contenidos a clasificar”, conlleva a que los documentos se inserten en el repositorio documental carentes de una supervisión de clasificación.

Debido al empleo de esta clasificación, los documentos no están organizados por la temática que abarcan. Esto conlleva a que el sistema de búsqueda en la actualidad este implementado para localizar los términos que introduce el usuario en el campo de búsqueda. Así, si el usuario desea encontrar, por ejemplo, los trabajos que se centran en el desarrollo de algoritmos, tendría que buscar en todos los estudios realizados que contengan la palabra algoritmo.

Esto supone que al realizar una indagación en el catálogo, no se obtenga toda la información del tema en cuestión, o que se obtengan varios documentos en los que realmente no se abarca la materia abordada, debido principalmente a una incorrecta y arbitraria clasificación de documentos.

Estas limitantes dificultan la interacción del usuario con el portal de la biblioteca pues implican retrasos innecesarios en el proceso de selección y adquisición de materiales, provocando que se afecte la navegabilidad en el sistema de información de la biblioteca.

De la problemática antes presentada emana el siguiente **problema de investigación:** La clasificación conceptual y no temática en el repositorio documental de la Universidad de las Ciencias Informáticas afecta la navegabilidad en el sistema de información de la biblioteca.

Para darle solución al problema se define como **objetivo general**, implementar una clasificación supervisada de documentos, mediante el modelo gráfico probabilístico Naive Bayes en el repositorio documental de la Universidad de las Ciencias Informáticas, para contribuir a mejorar la navegabilidad.

Determinándose como **objeto de investigación,** los clasificadores supervisados de documentos basados en modelos gráficos probabilísticos y como **campo de acción,** la clasificación supervisada de documentos.

Para poder dar cumplimiento al objetivo, se plantearon los siguientes **objetivos específicos**:

- Revisar el estado del arte sobre clasificación supervisada, en sistemas informáticos con el fin de sentar las bases de la investigación y el desarrollo del clasificador. Para ello, se deben realizar búsquedas en Internet y consultar los recursos del repositorio institucional de la Universidad de las Ciencias Informáticas así como analizar las herramientas que pueden ser utilizadas en el desarrollo del sistema.
- Analizar y diseñar el algoritmo de clasificación para guiar el proceso de desarrollo. Para alcanzar este objetivo, se debe analizar el modelo Naive Bayes, definir los requerimientos del sistema y generar los artefactos necesarios.
- Analizar cómo tratar los documentos con la perspectiva de una correcta selección de atributos para aplicar el modelo de clasificación. En la persecución de este objetivo, se deben investigar vías que faciliten la eliminación de palabras que no aporten información al clasificador así como los signos de puntuación.
- Seleccionar la colección de entrenamiento, para lograr que el clasificador brinde resultados positivos. Para lograrlo, se buscará en Internet información acerca de este tipo de colecciones.
- Desarrollar el modelo bayesiano de clasificación para solucionar el problema que genera investigación. Para alcanzar este objetivo, se necesita implementar funciones que permitan realizar los cálculos probabilísticos en los cuales se basa el modelo.
- Evaluar el algoritmo clasificador, para determinar si es factible utilizarlo. Para ello, se requiere realizar pruebas al código y pruebas de precisión.

Después de haber detallado los objetivos específicos; se identifica un grupo de **tareas,** encaminadas a resolver los dilemas particulares de la problemática planteada:

- Revisión de estudios enfocados en la clasificación supervisada, haciendo énfasis en los clasificadores basados en modelos bayesianos con el propósito de adquirir conocimientos básicos de este campo.

- Partiendo del estado del arte del tema, confeccionar el marco teórico que permita comprender sus principales características, además de conocer cómo y para qué se han empleado.
- Selección de las herramientas, lenguajes y metodologías con el fin de utilizar las más adecuadas para el desarrollo del clasificador.
- Realizar el análisis y diseño del modelo probabilístico que permita clasificar los documentos en la clase correspondiente.
- Selección de una métrica de suavizado que permita disminuir la hipótesis de independencia condicional.
- seleccionar un método de eliminación de términos poco significativos así como signos de puntuación para lograr una correcta selección de atributos.
- Selección de la colección de entrenamiento que posibilite al clasificador brindar resultados positivos.
- Implementación del algoritmo que permita la clasificación temática de los documentos.
- Selección de las colecciones que permitan realizar las pruebas de precisión.
- Realizar pruebas de Caja Blanca al código para evaluar su funcionamiento.
- Realizar las pruebas en términos de precisión al algoritmo para evaluar los resultados del clasificador.

Métodos de investigación:

El vocablo método, proviene de las raíces: meth, que significa meta, y odos, que significa vía. Por tanto, el método de investigación es la vía para llegar a la meta o final del estudio realizado.

En la tesis se usarán los siguientes métodos de investigación:

Métodos teóricos:

Histórico-lógico: Para analizar la trayectoria de los clasificadores supervisados de documentos y conocer así su funcionamiento y desarrollo.

Analítico-sintético: Para el análisis del funcionamiento de los clasificadores de documentos basados en modelos gráficos probabilísticos, y obtener el conocimiento necesario para la implementación de una solución para el problema planteado.

Métodos empíricos:

Observación: Para obtener la información relacionada con los clasificadores de documentos y conocer los aspectos generales del funcionamiento de estas aplicaciones así como las características fundamentales de los sistemas a estudiar.

Revisión de Documentos: Para lograr un mayor conocimiento del proceso de clasificación, lograr una mejor comprensión del desempeño y funcionamiento de clasificadores de documentos que existen actualmente y la manera en que operan, así como para determinar la forma de trabajo.

Para una mejor organización del contenido, la tesis se estructuró de la siguiente forma:

Capítulo 1: Estudio sobre clasificación supervisada de documentos, en sistemas informáticos: Se realiza un estudio de los elementos y conceptos relacionados con la clasificación documental. Se plasma la evolución que ha tenido el algoritmo que se implementará así como los estudios que se han realizado de este algoritmo en diversas partes del mundo incluyendo nuestro país. Además se especifican las herramientas a utilizar durante el desarrollo de la solución propuesta.

Capítulo 2: Análisis del clasificador de documentos basado en el modelo gráfico probabilístico Naive Bayes: Profundiza en las características generales y especificas del clasificador de documentos seleccionado. Se describe la solución propuesta, definiéndose los requerimientos a implementar. Además se describen elementos relacionados con el análisis y diseño de la solución.

Capítulo 3: Implementación y pruebas del clasificador de documentos mediante el modelo Naive Bayes: Describe los componentes fundamentales de la implementación. De igual forma se describe el proceso de pruebas del algoritmo a implementar, el cual incluye el diseño y aplicación de las pruebas.

Capítulo 1 Estudio sobre clasificación supervisada de documentos, en sistemas informáticos.

1.1 Introducción

Para comprender en profundidad los clasificadores de documentos, se hace necesario el estudio de los mecanismos y herramientas utilizadas para dicha función. En este capítulo se presentará un resumen de las tecnologías, metodología y herramientas seleccionadas para implementar la solución propuesta. Se realizará un estudio de los clasificadores de documentos en Cuba y en el mundo, en busca de reunir suficientes elementos para desarrollar un clasificador que cumpla con las necesidades de sus usuarios. Además, se hará énfasis en varios conceptos necesarios para un mayor entendimiento de los elementos de la investigación.

1.2 Descripción de términos

Para un mejor entendimiento de la clasificación supervisada de documentos, se realizó un estudio de los principales conceptos presentes en la bibliografía. A continuación se exponen los términos más significativos para una mejor comprensión de los aspectos tratados en la investigación.

Clasificación Automatizada de texto

La Categorización Automatizada de Textos ha tenido un interesante auge en los últimos 10 años, debido a la creciente disponibilidad de documentos en formato digital y la consiguiente necesidad de organizarlos. El enfoque dominante para este problema se basa en técnicas de Aprendizaje Automático: un proceso general inductivo que construye automáticamente un categorizador mediante el aprendizaje de características de categorías, a partir de un conjunto de documentos pre categorizado. Las ventajas de este método sobre el enfoque de Ingeniería del Conocimiento son: muy buena efectividad (comparable con la conseguida por expertos humanos), el ahorro considerable en términos de la exigente labor del experto y la sencilla portabilidad a dominios diferentes(SEBASTIANI, 2002).

La clasificación de textos, consiste en asignar a cada documento de la colección una etiqueta, que designa a la clase a la que pertenece. La decisión sobre qué etiqueta debe asignarse a un

documento determinado, se toma a partir de un modelo construido con una parte de la colección, denominada conjunto de entrenamiento. El objetivo de la tarea, es construir un modelo que prediga correctamente la clase de un conjunto de documentos, llamado de prueba, que no intervinieron en la construcción del modelo (ÁLVAREZ, 2009).

Navegabilidad

La navegabilidad o navegabilidad web es la facilidad con la que un usuario puede desplazarse por todas las páginas que componen un sitio web. Para lograr este objetivo, un sitio web debe proporcionar un conjunto de recursos y estrategias de navegación diseñados para conseguir un resultado óptimo en la localización de la información y en la orientación para el usuario (GIULIANELLI *et al.*, 2008).

También entiéndase por navegabilidad a la posibilidad o facilidad de un agente del entorno de transitar desde un lugar hacia otro siguiendo ciertas reglas como el sentido y dirección de los caminos (CEDEÑO, 2010).
En resumen la navegabilidad puede definirse como la facilidad que brinda un sistema a los usuarios de recorrer todos los espacios a los cuales tiene acceso sin que esto se haga una tarea tediosa.

Precisión

Según la Real Academia Española (RAE), se hace referencia a precisión como adjetivo de un aparato, de una máquina, de un instrumento, etc., cuando es construido con singular esmero para obtener resultados exactos (RAE, 2001).

La precisión estima la probabilidad de que una etiqueta proporcionada por el modelo es correcta (GOUTTE y GAUSSIER, 2005).
Es la cercanía con la que sucesivas observaciones se ajustan a sí mismas (ORTEGA, 2008).

Se refiere a la capacidad de un instrumento de dar el mismo resultado en mediciones diferentes, realizadas en las mismas condiciones. No tienen nada que ver con la relación con un valor real (ORTEGA, 2008).

Precisión se refiere a la dispersión del conjunto de valores obtenidos de mediciones repetidas de una magnitud. Cuanto menor es la dispersión mayor la precisión. Puede definirse como la

proximidad de concordancia entre valores medidos, obtenida por mediciones repetidas de un mismo objeto, o de objetos similares bajo condiciones especificadas (CASTELLANO, 2009).

La precisión es un término relacionado con la confiabilidad de un instrumento, es decir, si un instrumento proporciona resultados similares cuando se mide un material de referencia de manera repetida, entonces el instrumento es preciso (RAMÍREZ, 2009).

De manera general la precisión es el equilibrio existente entre varios experimentos. En el caso específico de la categorización de textos pudiera definirse como, el grado en que un documento puede ser clasificado correctamente. Más adelante se explica cómo será calculada la presión del algoritmo, dando como resultado un valor en el rango de 0 a 1 por lo que mientras más cercano sea el valor a 1, más preciso será el clasificador.

Modelos gráficos probabilísticos

Los modelos gráficos probabilísticos unen las posibilidades de expresión de los grafos en procesos de razonamiento complejos, y toda la teoría de la probabilidad ampliamente integrada en el campo científico.

Los modelos gráficos probabilísticos, representan distribuciones de probabilidad conjuntas multivariante. Esta representación se expresa mediante un conjunto de términos, cada uno de los cuales abarca solo unas pocas variables de todo el conjunto de variables consideradas. La estructura del producto se representa mediante un grafo que relaciona variables que aparecen en un mismo término.

Es el grafo el que especifica la forma del producto de las distribuciones y provee, así mismo, de herramientas para el razonamiento sobre propiedades vinculadas a los términos de dicho producto (LAURITZEN y SPIEGELHALTER, 1988).

Clasificación supervisada

En la clasificación supervisada o categorización, se parte de una serie de clases o categorías conceptuales prediseñadas. La labor del clasificador es asignar cada documento a la clase o categoría que le corresponda (SOUMEN, 2003). Existen aplicaciones de la clasificación supervisada en áreas del conocimiento tan diversas como la clasificación de texto (COHEN y HUNTER, 2008), la ecología(GIBERT *et al.*, 2008) y la genómica(YANG, *et al.*, 2009) .

Aunque hay muchos algoritmos capaces de lograr una clasificación supervisada, la idea central es muy similar ya que consiste en construir un patrón propio para cada clase o categoría y aplicar alguna función para estimar la similitud entre el documento a clasificar y cada uno de los patrones de las categorías (HE *et al.*, 2003; VILLA y ALFONSO, 2005; WITTEN y EIBE, 2005).

1.3 Algoritmos de Clasificación Supervisada

Para lograr una Clasificación Supervisada de Textos se debe disponer de una Colección de Entrenamiento que contenga los rasgos distintivos de cada clase. Cuanto mayor sea el conjunto de documentos pre-categorizados, mayor será la información característica de cada clase y, por tal motivo, mejor resultará el aprendizaje. Por su parte, los sistemas de Clasificación No Supervisada son aquellos que no disponen de conocimiento previo, es decir, de una Colección de Entrenamiento.

El aprendizaje o entrenamiento es la etapa en la que se obtiene la información para cada una de las categorías en las que será posible categorizar un documento determinado. La obtención de las características de categorías es el resultado de la etapa de entrenamiento. Este entrenamiento se realiza sobre una jerarquía temática que puede estar estructurada en uno o varios niveles de generalidad. A partir de un número de documentos que resulte representativo de cada categoría, es posible encontrar las características de esta (YANG, 1999). Luego de realizar el entrenamiento la Colección de Entrenamiento está en condiciones de ser utilizada por los algoritmos de Clasificación Supervisada.

Existen varios algoritmos dedicados a la Clasificación Supervisada pero, es válido destacar que muchos no han sido solamente utilizados en la Clasificación de Textos, sino que se utilizan para clasificar diversos objetos. Entre los más utilizados se citan:

Algoritmo de Rocchio

Este algoritmo proporciona un mecanismo para construir vectores representativos (patrones) de cada una de las categorías de documentos que se consideren. De este modo, en la fase de entrenamiento se parte de una Colección de Entrenamiento pre-categorizada manualmente y se construyen patrones para cada una de las categorías, considerando como ejemplos positivos los documentos de entrenamiento de esa categoría y como ejemplos negativos los del resto. Para

categorizar un nuevo documento, bastará con encontrar la distancia entre la representación de los documentos y cada uno de los patrones. Aquel patrón que presente mayor similitud indicará la categoría a la que se debe asignar el documento (RIPOLL y PÉREZ, 2008).

Algoritmo de los Vecinos más Cercanos y variantes

Este algoritmo se basa en la aplicación de una métrica que establezca la similitud entre un documento que se quiere categorizar y cada uno de los documentos de la Colección de Entrenamiento. La categoría que se asigna al documento será la del documento más cercano según la métrica establecida. Una de las variantes más conocidas de este algoritmo es la de los *k*-vecinos más cercanos, que consiste en tomar los *k* documentos más parecidos en lugar de sólo el primero. Como los documentos más cercanos pueden pertenecer a categorías diferentes, se asignará aquella que más veces haya aparecido (BERMEJO, 2000).

Algoritmos basados en redes neuronales

Las redes neuronales han sido aplicadas a problemas de categorización de documentos en numerosas ocasiones. Es posible entrenar una red neuronal para que dada una entrada determinada (un vector de representación) produzca una salida deseada (la categoría a la que corresponde el documento). Existen muchos tipos de redes neuronales, con topologías y características bien diferenciadas (FRESNO, 2006).

Algoritmo Naive Bayes

Naive Bayes es una técnica de clasificación y predicción que construye modelos que predicen la probabilidad de posibles resultados. Naive Bayes utiliza datos históricos para encontrar asociaciones y relaciones y hacer predicciones. Este algoritmo predice resultados binarios o multiclase. En los problemas binarios, cada registro cumplirá o no el comportamiento modelado. Naive Bayes puede hacer predicciones para problemas multiclase, en los cuales hay varios resultados posibles (WITTEN y EIBE, 2005).

Se basa en la teoría de probabilidad de Bayes. El teorema de Bayes permite estimar la probabilidad de un suceso a partir de la probabilidad de que ocurra otro, del cual depende el primero. Este algoritmo es el más conocido dentro de los probabilísticos (FRESNO, 2006).

Ha sido objeto de estudio de varios investigadores (KONONENKO, 1991; PAZZANI,1996; SAHAMI, 1996; PAZZANI, 1998; KEOGH y PAZZANI, 1999; FIGUEROA *et al.*, 2005) y ha sido utilizado en varios trabajos relacionados con el tema (DUDA y HART, 1973; CESTNIK *et al.*, 1987; ARMAÑANZAS, 2004; ORNELLA, 2010; GÓMEZ *et al.*, 2011).

Luego de analizar varios de estos estudios en los que se incluyen experimentos con varios algoritmos (incluyendo Naive Bayes), se aprecia que la mayoría de los algoritmos brindan resultados similares pero, según la comparación realizada por Balbontín y Sánchez (2004) entre los algoritmos C4.5, PART y Naive Bayes, C4.5 y PART tardan mucho más en mostrar los datos debido a que los cálculos que realizan para obtener dichos datos son más complejos. Además según Suárez (2005) el problema de la resolución de la ambigüedad semántica de las palabras, que es uno de los problemas frecuentes de los clasificadores automatizados no radica tanto en los métodos utilizados como en los datos que manejamos, los ejemplos de entrenamiento y los textos donde se aplican, finalmente, para su clasificación.

Por estas razones será el algoritmo Naive Bayes el utilizado para el desarrollo del clasificador.

1.4 Algoritmo Naive Bayes y su evolución

Dentro de los algoritmos probabilísticos bien conocidos, se encuentra el denominado Naive Bayes. Su labor se centra en estimar la probabilidad de que un documento pertenezca a una categoría. Se basa en el supuesto de independencia condicional entre los atributos predictores dado el valor de la clase, en que los atributos numéricos siguen una distribución normal y en que no existen variables ocultas que influyan en el proceso de predicción. Para que un documento sea clasificado en una clase, debe cumplir con las características definidas a priori para esta categoría.

Estas características suelen ser los términos que conforman los documentos y tanto su probabilidad de aparición general, como la probabilidad de que aparezcan en los archivos de una determinada categoría, pueden obtenerse a partir de la frecuencia de aparición en la colección de entrenamiento.

Es válido destacar que si las colecciones de entrenamiento son pequeñas, pueden ocurrir errores al estimar las probabilidades. Un ejemplo frecuente, está presente cuando un determinado término no aparece en la colección de entrenamiento, pero aparece en los documentos a clasificar. Esto se soluciona aplicando técnicas de suavizado para evitar fallos a la hora de obtener probabilidades (FIGUEROA *et al.*, 2005).

Los investigadores del aprendizaje automático, se han dado cuenta de su potencialidad y robustez en problemas de clasificación supervisada, por lo que han encaminado sus estudios a perfeccionarlo para lograr mejores resultados. Esto se evidencia en el estudio realizado por Pedro Larrañaga, Iñaki Inza, Abdelmalik Moujahid (2007) en el que plantean que:

Kononenko (1991) introduce el denominado seminaive Bayesian classifier. En el mismo se trata de evitar las estrictas premisas sobre las que se construye el paradigma Naive Bayes por medio de la consideración de nuevas variables en las cuales no necesariamente tenga que aparecer el producto cartesiano de dos variables, sino tan sólo aquellos valores de dicho producto cartesiano que verifiquen una determinada condición que surge al considerar el concepto de independencia junto con el de la habilidad en la estimación de las probabilidades condicionadas, cuestión esta última que es resuelta a partir del teorema de Chebyshev.

Sahami (1996) presenta un algoritmo denominado *k Dependence Bayesian classifier* [kDB por sus siglas en ingles), el cual posibilita atravesar el amplio espectro de dependencias disponibles entre el modelo Naive Bayes y el modelo correspondiente a una red bayesiana completa. El algoritmo se fundamenta en el concepto de clasificador Bayesiano k-dependiente, el cual contiene la estructura del clasificador Naive Bayes y permite a cada uno de los términos predictores tener un máximo de k variables padres sin contar a la variable clase.

Pazzani (1998) introduce el concepto de inducción constructiva con el que a partir del producto cartesiano entre variables y usando el algoritmo voraz *Backward Sequential Elimination and Joining* (BSEJ por sus siglas en inglés) de una manera de envoltura, desarrolla modelos de clasificadores Naive Bayes así como K-NN (*K nearest neighbours*, vecinos más cercanos).

Posteriormente Pazzani (1996) presenta una aproximación, en la que de manera voraz se va construyendo un modelo Naive Bayes en el que se detectan aquellas variables irrelevantes, así

como aquellas variables dependientes entre sí. Cuando se detectan variables dependientes, se crea una nueva variable a partir del producto cartesiano de las mismas.

El algoritmo, está guiado por un score que resulta ser la validación del porcentaje de bien clasificados. Se presentan dos algoritmos voraces, uno hacia adelante denominado *Forward Sequential Selection and Joining* (FSSJ por sus siglas en inglés) y otro hacia atrás Backward Sequential Elimination and Joining (BSEJ por sus siglas en inglés).

Friedman y col.(FRIEDMAN *et al.*, 1997) presentan un algoritmo denominado *Tree Augmented Network* (TAN) el cual consiste básicamente en una adaptación de un algoritmo creado por Chow y Liu (CHOW y LIU, 1968). En el mismo, se tiene en cuenta la cantidad de información mutua condicionada a la variable clase, en lugar de la cantidad de información mutua en la que se basa el algoritmo de Chow y Liu.

Keogh y Pazzani (1999) proponen un algoritmo voraz, que va añadiendo arcos a una estructura Naive Bayes. En cada paso se añade el arco que, manteniendo la condición de que en la estructura final cada variable no tenga más de un padre, mejore en mayor medida el porcentaje de bien clasificados obtenidos mediante el mismo.

1.5 Estudios relacionados al algoritmo

Para comprender dónde y cómo se ha usado en los últimos años el modelo Naive Bayes, se realizó un estudio que no solo se centrará en las aplicaciones que ha tenido en la clasificación de documentos, sino que abarcara otras áreas para comprobar su utilidad.

Algoritmo de selección de atributos basado en utilidad incremental y análisis de redundancia: Aplicación a datos biológicos.

Uno de los principales puntos de interés en Bioinformática es la selección de atributos en tareas de clasificación. La identificación y eliminación de parte de la información no sólo puede beneficiar en la capacidad predictiva del modelo; sino también en su eficiencia y comprensibilidad.

Para lograr una eficiente selección de atributos, varios profesionales de la Universidad Pablo de Olavide y la Universidad de La Laguna en Tenerife, ambas de España, realizaron un estudio para

proponer un algoritmo de selección que reduzca significativamente el subconjunto de atributos degradar la capacidad predictiva(GÓMEZ *et al.*, 2011).

Como parte de esa investigación elaboraron un experimento con objetivo de evaluar el algoritmo propuesto. Los clasificadores usados en los experimentos, debido a su popularidad, fueron el Naive Bayes y el Support Vector Machine (SVM por sus siglas en inglés).

Medidas de filtrado de selección de variables mediante la plataforma "Elvira".

En agosto del 2004, el alumno Rubén Armañanzas Arnedillo, teniendo como tutor a Iñaki Inza Cano desarrolló el proyecto de fin de la carrera de Ingeniería en Informática, titulado Medidas de filtrado de selección de variables mediante la plataforma “Elvira”. El objetivo del proyecto consistía en el estudio, implementación y validación en bases de datos reales, de varias medidas de filtrado para la selección de variables en problemas de clasificación supervisada.

Para cumplir con dicha tarea el estudiante utilizó varias técnicas basadas en redes bayesianas. Dentro de ellas sobresalen las referentes a Naive Bayes, un ejemplo de esto es el paquete de clases Elvira.learning.classification.supervised.discrete la cual contiene las clases necesarias para aprender modelos clasificatorios bayesianos supervisados y donde destacan los clasificadores *Naive Bayes, Selective Naive Bayes, Semi Naive Bayes y tree augmented Naive Bayes* (ARMAÑANZAS, 2004).

Naive Bayes Multinomial para Clasificación de Texto Usando un Esquema de Pesado por Clases.

En abril de 2009, Emmanuel Anguiano Hernández desarrolló una investigación, con el propósito de mejorar el desempeño de un clasificador Naive Bayes. Para ello se empleó un modelo probabilista, particularmente Naive Bayes Multinomial para clasificar documentos de una colección de noticias con una representación vectorial y un esquema de pesado que considera la frecuencia de aparición de cada término del vocabulario en el documento para el entrenamiento del modelo. Además la representación de los documentos de prueba se modifica usando un esquema de pesado por clases.

En este trabajo se utilizó la colección R8 de documentos previamente clasificados para entrenar y probar el sistema. R8 es una sub-colección de Reuters-21578 que a su vez, es una colección de noticias de la agencia Reuters del año 1987 que son usadas como un estándar para evaluar

sistemas. Se eligió R8 ya que presenta gran desbalance en el número de noticias que pertenecen a cada una de las clases y por lo tanto es adecuada para probar si un sistema de clasificación es eficiente (ANGUIANO, 2009).

Implementación de Redes bayesianas mediante una aplicación para la detección y el filtrado de SPAM.

A medida que Internet se expande, el número de usuarios de esta red mundial crece drásticamente. Junto con este crecimiento y la utilización de este medio de comunicación ha surgido la utilización del email como medio de difusión publicitario para hacer llegar en masa avisos publicitarios a potenciales interesados. Estos mails no solicitados fueron creciendo exponencialmente a medida que pasa el tiempo llegando a ser un problema y recibieron la categorización de mails basura o SPAM.

En agosto de 2006, Esteban Calabria que en ese período se desempeñaba como ayudante de la materia de Algoritmos y Programación I en la Universidad de Buenos Aires, realizó un trabajo investigativo con el propósito de desarrollar la temática del aprendizaje activo supervisado para resolver el problema de la clasificación utilizando métodos de decisión basados en estadística y la teoría de decisión bayesiana como el clasificador Naive Bayes. Además de mostrar las ventajas que nos ofrece este aprendizaje activo en comparación con el aprendizaje pasivo y una posible aplicación para el filtrado del correo basura (CALABRIA, 2006).

Propuesta de utilización de Redes Bayesianas Naive para la detección de anomalías en el tráfico Ethernet.

Con el objetivo de determinar anomalías en el tráfico Ethernet a partir de un grupo de variables entre las cuales se encuentran los puertos y direcciones de origen y destino, los protocolos utilizados, etc. los ingenieros cubanos Oscar Méndez González, del Organismo Central del Ministerio de Educación Superior (MES) y Yasser Aquino Rivera, de la Dirección Nacional de los Joven Club de Computación y Electrónica propusieron en el año 2011 utilizar un algoritmo de clasificación.

Méndez y Aquino, hacen referencia al uso de Redes Bayesianas de tipo Naive orientadas a la detección de anomalías en el tráfico de redes y los beneficios que pudiese brindar una solución

óptima a la problemática de los falsos positivos, así como el exagerado número de alertas que generan estas aplicaciones (MÉNDEZ y AQUINO, 2011).

1.6 Tecnologías necesarias para la implementación del algoritmo.

A partir del análisis de la bibliografía y teniendo en cuenta el entorno donde se utilizará el algoritmo, se determinó que para la realización del nuevo sistema que satisfaga las necesidades específicas del cliente, es recomendable la utilización de herramientas de software, lenguajes de programación y de modelado así como una metodología que guie el proceso de desarrollo.

1.6.1 Lenguaje Unificado de Modelado 2.1

El Lenguaje Unificado de Modelado (UML por sus siglas en ingles), es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. UML no es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso. Es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos (SALINAS y HISTCHFELD, 1999).

Según Hernández (2008), los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- Visualizar: UML permite expresar gráficamente un sistema de manera que otro lo pueda entender.
- Especificar: UML permite especificar cuáles son las características de un sistema antes de su construcción.
- Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.
- Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden ser aprovechados para su futura revisión.

El UML es perfecto para el modelado de sistemas orientados a objetos ya que incluye la representación de abstracciones, herencias, polimorfismos, encapsulamientos, envío de mensajes, asociaciones y agregaciones. Permite detectar con facilidad las dependencias y

dificultades implícitas del sistema. Se pueden modelar tanto sistemas de software como de hardware.

Existen otros Lenguajes de Modelado como el Lenguaje de Modelado Conceptual (ConML) pero este está enfocado al modelado conceptual, no así el UML que se centra en el modelado de software Debido a las características que posee, se utilizará el UML para el modelado del sistema que se desarrollará.

1.6.2 Metodologías de desarrollo de software

En un proceso de desarrollo de software puede ser muy necesario el uso de alguna metodología ya que generalmente cuando se desea realizar alguna tarea con la calidad requerida es imprescindible tener una guía por la cual regirse.

Para seleccionar la metodología a utilizar, se analizaron varias de las existentes en el mundo. Además, teniendo en cuenta que se desea desarrollar un algoritmo, proceso para el cual no se requiere que se generen todos los artefactos necesarios para proyectos mayores, se decide utilizar XP ya que esta es una metodología ágil. A continuación se exponen varias características de la misma.

Metodología XP

De las metodologías ágiles, XP es la más popular en la actualidad. Es una metodología centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Esta metodología se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, en la comunicación fluida entre todos los participantes, la simplicidad en las soluciones implementadas y el coraje para enfrentar los cambios. Se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes(LASCANO *et al.*, 2011).

Las prácticas de esta metodología que se utilizan en esta investigación se exponen a continuación:

Planificación: En esta práctica el equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración estableciendo la prioridad de cada historia de usuario, de acuerdo con el valor que aporta para el negocio. Los programadores estiman el esfuerzo asociado a cada historia de usuario. Se ordenan las historias de usuario según prioridad y esfuerzo, y se define el contenido de la entrega y/o iteración, apostando por enfrentar lo de más valor y riesgo cuanto antes (BECK, 1999).

Refactorización: La refactorización es una actividad constante de reestructuración del código con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios. La refactorización mejora la estructura interna del código sin alterar su comportamiento externo (STEPHENS y ROSENBERG, 2004).

Estándares de programación: XP enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación (del equipo, de la organización u otros estándares reconocidos para los lenguajes de programación utilizados). Los estándares de programación mantienen el código legible para los miembros del equipo, facilitando los cambios (JEFFRIES *et al.*, 2001).

1.6.3 Lenguaje de programación

Un lenguaje de programación es aquel elemento dentro de la informática que nos permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes (DEFINICIÓN, SF).

Lenguaje de programación PHP 5.3.8

PHP, acrónimo de *Hypertext Preprocessor*, es un lenguaje de programación interpretado del lado del servidor, el cual fue diseñado para el desarrollo de sitios web dinámicos. Es un lenguaje relativamente fácil de usar, rápido e integrable, que es introducido dentro del código HTML, permite el uso de técnicas de programación orientada a objeto, biblioteca nativa de funciones sumamente amplia e incluida, no requiere definición de tipos de variables y tiene manejo de excepciones. Es un lenguaje multiplataforma por lo que puede ser usado en diferentes sistemas operativos, al igual que con diferentes servidores web. Por cumplir la condición de ser código

abierto, cuenta con el apoyo de un gran grupo de programadores para corregir errores, además de estar actualizándose continuamente con mejoras para ampliar las capacidades del mismo.

Una de sus características más potentes es que soporta varios gestores de base de datos, como MySQL, Oracle y PostgreSQL.

PHP es un lenguaje de scripting muy usado, es especialmente adecuado para el desarrollo web y puede ser embebido en páginas HTML

Debido a su amplia distribución PHP está perfectamente soportado por una gran comunidad de desarrolladores (MORENO y VANEGAS, 2009).

Por las características antes expuestas y por ajustarse a la futura integración con el portal de la biblioteca, el cual se encuentra desarrollado con el Sistema de Administración de Contenidos (CMS por sus siglas en ingles), fue seleccionado PHP como lenguaje para la implementación del clasificador de documentos.

1.6.4 Herramientas de desarrollo

Las herramientas de desarrollo pueden considerarse aquellos programas o aplicaciones informáticas que permiten que el desarrollo de un programa sea lo más factible posible.

Entorno de desarrollo integrado NetBeans 7.2

Es un proyecto exitoso de código abierto con una gran base de usuarios, una comunidad en constante crecimiento. Sun MicroSystems[1] fundó este proyecto en junio 2000 y continúa siendo su patrocinador principal. En la actualidad el proyecto cuenta con dos productos disponibles: el Entorno de desarrollo integrado y la plataforma.

El entorno de desarrollo es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Es un producto libre y gratuito sin restricciones de uso.

[1] Empresa informática comprada por la Corporación Oracle antes era parte de Silicon Valley, fabricante de semiconductores y software.

Esta herramienta permite desarrollar múltiples proyectos en disímiles lenguajes de programación como son: Java, PHP y otros. Posee una amplia documentación, facilidades de auto-completamiento y generación automática de código. Así como también incluye muchas librerías y pluings para varios lenguajes de programación (SITIO OFICIAL DE NETBEANS, 2012).

Por las características que posee este Entorno de Desarrollo Integrado (IDE) y principalmente por las facilidades del auto-completamiento y el señalamiento de errores se escoge para escribir el código del clasificador de documentos basado en el modelo Naive Bayes.

1.6.5 Herramientas CASE

Acrónimo de Computer Aided Software Engineering (Ingeniería de Software Asistida por Computadoras). Las herramientas CASE son un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software (MURILLO, 2008).

Visual Paradigm for UML 8.0

Es una herramienta CASE profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite realizar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Una de sus características más importantes es que es multiplataforma (GONZÁLEZ y ROMERO, 2012).

Visual Paradigm como herramienta CASE ofrece:

- Entorno de creación de diagramas para UML 2.1.
- Diseño centrado en casos de uso y enfocado al negocio que permite generar un software de mayor calidad.
- Uso de un lenguaje estándar, común para todo el equipo de desarrollo, lo cual facilita la comunicación.
- Capacidades de ingeniería directa e inversa.
- Modelo y código permanecen sincronizados durante todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.

- Disponibilidad de integrarse a los principales IDE's.
- Disponibilidad en múltiples plataformas tanto en sistemas operativos Windows como en las distribuciones de GNU\Linux"

Para seleccionar la herramienta CASE adecuada se revisaron también las características de Rational Rose pero por no contar con una versión para Sistemas Operativos Linux fue desechada. Es posible concluir entonces que el Visual Paradigm es el más apropiado para el modelado del clasificador. Esta herramienta permite el modelado orientado a objetos y basado en el Lenguaje Unificado de Modelado; es multiplataforma y posee gran disponibilidad de integración con varios entornos de desarrollos integrados que están bajo licencias de software libre como NetBeans.

1.7 Conclusiones parciales

Luego de analizar los temas referentes a los clasificadores supervisados se logra comprender los conceptos fundamentales y el estado del arte de estos sistemas. Después de analizar varias herramientas y tecnologías, se seleccionan cuáles son las más adecuadas para el desarrollo del sistema, dentro de las cuales se encuentran: el lenguaje de modelado UML versión 2.1 y el lenguaje de programación PHP en su versión 5.3.8. Las herramientas que se recomiendan son Netbeans version 7.2 para la generación de código fuente y Visual Paradigm en la versión 8.0 para el modelaje del sistema.

Capítulo 2: Análisis del clasificador de documentos basado en el modelo gráfico probabilístico Naive Bayes.

2.1 Introducción

En este capítulo se explicará la propuesta de solución a la situación problemática, abordando cada uno de los requerimientos de un método de clasificación. Además se especificarán las funcionalidades del clasificador que se desarrollará. Se incluirán los artefactos generados y se detallarán los elementos de análisis y diseño necesarios para la construcción del algoritmo.

2.2 Propuesta de solución

La clasificación de documentos digitales, debido al significativo incremento de datos, se ha convertido en una tarea sumamente importante para la recuperación de información. Con el objetivo de lograr una clasificación eficiente son varios los métodos utilizados en el mundo. En el período comprendido entre las últimas décadas del siglo XX y la primera del siglo XXI se ha evidenciado un interés creciente en la utilización de modelos gráficos probabilísticos para clasificación. Éstos han demostrado acomodarse a la naturaleza flexible de numerosos conceptos y además, gozan de una sólida base en la teoría de la probabilidad.

El método gráfico probabilístico para clasificación conocido como Naive Bayes se basa en la aplicación del teorema de Bayes.

$$p(A|B) = \frac{P(A)*P(B|A)}{P(B)} \qquad (1)$$

donde:

- $p(A)$ es conocido como la probabilidad a priori de que el suceso A sea cierto.
- $p(A|B)$ es conocido como la probabilidad a posteriori, o, la probabilidad de que el suceso A sea cierto tras considerar B.
- $p(B|A)$ es conocido como verosimilitud, e indica la probabilidad de que el suceso B sea cierto, asumiendo que A lo es.

- $p(B)$ es la probabilidad a priori de que el suceso B sea cierto. Actúa de coeficiente normalizador en la fracción.

En la teoría de la probabilidad el Teorema de Bayes es un resultado enunciado por Thomas Bayes en 1763 que como se aprecia en la fórmula (1) expresa la probabilidad condicional de un evento aleatorio A dado B.

Un clasificador Naive Bayes o Bayesiano ingenuo como también se le conoce, es un clasificador probabilístico fundamentado en el teorema de Bayes y algunas hipótesis simplificadoras adicionales. Es a causa de estas simplificaciones, que se suelen resumir en la hipótesis de independencia entre los términos predictores, que recibe el apelativo de ingenuo.

Naive Bayes se puede considerar como una familia de algoritmos debido a que se ha modificado para mejorar su rendimiento. En esta familia se encuentran entre otros algoritmos el Naive Bayes Simple y el Naive Bayes Multinomial.

El clasificador Naive Bayes Simple considera la probabilidad de aparición de cada término dada la clase de forma binaria, es decir el término aparece o no y entonces su probabilidad condicional dada la clase es o no considerada. En este sentido, el clasificador Naive Bayes Multinomial suele mejorar el desempeño pues considera el número de apariciones del término para evaluar la contribución de la probabilidad condicional dada la clase con lo que el modelado de cada documento se ajusta mejor a la clase a la que pertenece. Es por esta característica que se decide implementar el algoritmo Naive Bayes Multinomial para darle solución a la problemática planteada.

El algoritmo se basa en la aplicación de la Regla de Bayes para predecir la probabilidad condicional de que un documento pertenezca a una clase $p(c_i|d_j)$ a partir de la probabilidad de los documentos dada la clase $p(d_j|c_i)$ y la probabilidad a priori de la clase en el conjunto de entrenamiento $p(c_i)$:

$$p(c_i|d_j) = p(c_i)p(d_j|c_i)/p(d_j) \qquad (2)$$

donde:

- d_j es el documento que se quiere clasificar.
- c_i es la clase para la cual se está verificando la pertenencia del documento d_j.

Dado que la probabilidad de cada documento $p(d_j)$ no aporta información para la clasificación, el término suele omitirse.

La probabilidad de un documento dada la clase suele asumirse como la probabilidad conjunta de los términos que aparecen en dichos documentos dada la clase y se calculan como:

$$p(d_j|c_i) = \prod_{t_k \in d_j} p(t_k|c_i)^{n_{jk}}, \qquad (3)$$

donde:

- t_k son las palabras distintas en el documento d_j.
- n_{jk} es el número de veces que la palabra t_k aparece en el documento d_j.

El término $p(t_k|c_i)$ se calcula a partir del número de apariciones de cada término t_k en una clase c_i pero para evitar el problema de las probabilidades 0 se usa la estimación de Laplace:

$$p(t_k|c_i) = \frac{N_{ik}+1}{N_i+M} \qquad (4)$$

donde:

- N_{ik} es el número de veces que el término t_k aparece en los documentos de la clase c_i.
- N_i es el número total de palabras en los documentos de la clase c_i.
- M es el tamaño del vocabulario (número de palabras distintas en la colección de entrenamiento))

La estimación de las probabilidades a priori de las clases $p(c_i)$, se hace generalmente por máxima verosimilitud:

$$p(c_i) = \frac{N_{i,doc}}{N_{doc}} \qquad (5)$$

Dónde:

- $N_{i,doc}$ es el número de documentos en la colección de entrenamiento que se le asigna a la clase c_i.

- N_{doc} es el número de documentos en la colección de entrenamiento.

2.3 Selección de una técnica de suavizado

Uno de los problemas que Naive Bayes presenta, es que asume fuertemente la independencia condicional entre los términos. Para eliminar esta debilidad, se asumió una técnica descrita por Guo Qiang de la Universidad de Ciencias de Ingenierías de Shanghai.

En el marco de la Segunda Conferencia Internacional sobre Investigación y Desarrollo de la Informática, Quian (2010) propone una mejora para disminuir la hipótesis de independencia de Naive Bayes para la clasificación de texto. En este lugar Quian planteaba: “El modelo multinomial Naive Bayes trata cada aparición de una palabra independiente de otra presencia de esa misma palabra. Sin embargo múltiples repeticiones de la misma palabra pueden no ser independientes. Cuando una palabra aparece es muy probable que se produzca de nuevo. El modelo multinomial no tiene en cuenta ese fenómeno. Múltiples ocurrencias ocupan porcentajes que aportan muchos datos irrelevantes. Esto resulta una gran subestimación de la probabilidad de documentos con múltiples ocurrencias de la misma palabra”.

Debido a este inconveniente Quian planteó una modificación para el modelo multinomial Naive Bayes. El propuso sustituir n_{jk} en (3) por (3.1):

$$min\{1 + log_2^{n_{jk}}, n_{jk}\} \qquad (6)$$

Esta técnica básicamente realiza una comparación entre $1 + \log_2^{n_{jk}}$ y n_{jk}, donde n_{jk} es la cantidad de apariciones del término en un documento, tomando de esta comparación el menor para utilizarlo en la clasificación.

Luego de ajustar la forma de contar las palabras, la probabilidad de la clase dado que se conoce el documento (3) quedaría de la siguiente forma:

$$p(d_j|c_i) = \prod_{t_k \in d_j} p(t_k|c_i)^{min(1+log_2^{n_{jk}}, n_{jk})} \qquad (3.1)$$

Esa es una de las múltiples formas que se han propuesto para disminuir la hipótesis de independencia condicional del modelo gráfico probabilístico Naive Bayes. Es válido destacar que aunque muchos han sido los intentos de reducir la independencia, según los estudios revisados en el primer capítulo, en la práctica no se ha eliminado del todo.

2.4 Pseudocódigos del algoritmo

El pseudocódigo se puede definir como un lenguaje entre el nuestro y el de las computadoras. En este apartado se presenta el pseudocódigo del algoritmo de clasificación que se propone.

Pseudocódigo para la clasificación

INICIO

Entrada: $D = \{d_1, d_2, \dots, d_j\}$ – Colección de documentos que se quiere clasificar.

$C = \{c_1, c_2, \dots, c_i\}$ – Conjunto de clases en las que se quiere clasificar los documentos.

Salida: DC – Documentos clasificados.

Para Cada $d_j \in D$ Hacer

Tratar el documento.

Aplicarle Naive Bayes.

Seleccionar la clase que mayor probabilidad obtuvo luego de aplicarle Naive Bayes.

Asignar el documento a la clase correspondiente.

FIN Para Cada

FIN

Pseudocódigo para Naive Bayes

INICIO

Entrada: d_j – Documento que se quiere clasificar.

$C = \{c_1, c_2, \dots, c_i\}$ – Conjunto de clases en las que se quiere clasificar los documentos.

Salida:($p(c_i|d_j)$)– Arreglo con las probabilidad de que conociendo el documento d_j pueda clasificarse en la clase c_i.

$p(c_i)$ – probabilidad a priori de la clase.

$p(d_j|c_i)$ – probabilidad de que el documento (d_j) pertenezca a una clase conocida (c_i).

$N_{i,doc}$ – número de documentos en la colección de entrenamiento que se le asigna a la clase c_i.

N_{doc} – número de documentos en la colección de entrenamiento.

N_{ik} – número de veces que el término t_k aparece en los documentos de la clase c_i.

N_i – número total de palabras en los documentos de la clase c_i.

M – tamaño del vocabulario (número de palabras distintas en la colección de entrenamiento).

t_k – palabras distintas en el documento d_j.

n_{jk} – número de veces que la palabra t_k aparece en el documento d_j.

Para Cada $c_i \in C$ Hacer

$$p(c_i|d_j) = p(c_i) * p(d_j|c_i)$$

$$p(c_i) = \frac{N_{i,doc}}{N_{doc}}$$

$$p(d_j|c_i) = \prod_{t_k \in d_j} p(t_k|c_i)^{min(1+log_2^{n_{jk}}, n_{jk})}$$

$$p(t_k|c_i) = \frac{N_{ik}+1}{N_i+M}$$

Escribir ($p(c_i|d_j)$)

FIN Para Cada

FIN

2.5 Modelo de dominio

A continuación se muestra el modelo de dominio que determina el entorno de la solución a implementar, donde se describen las clases que interactúan así como la asociación entre estas.

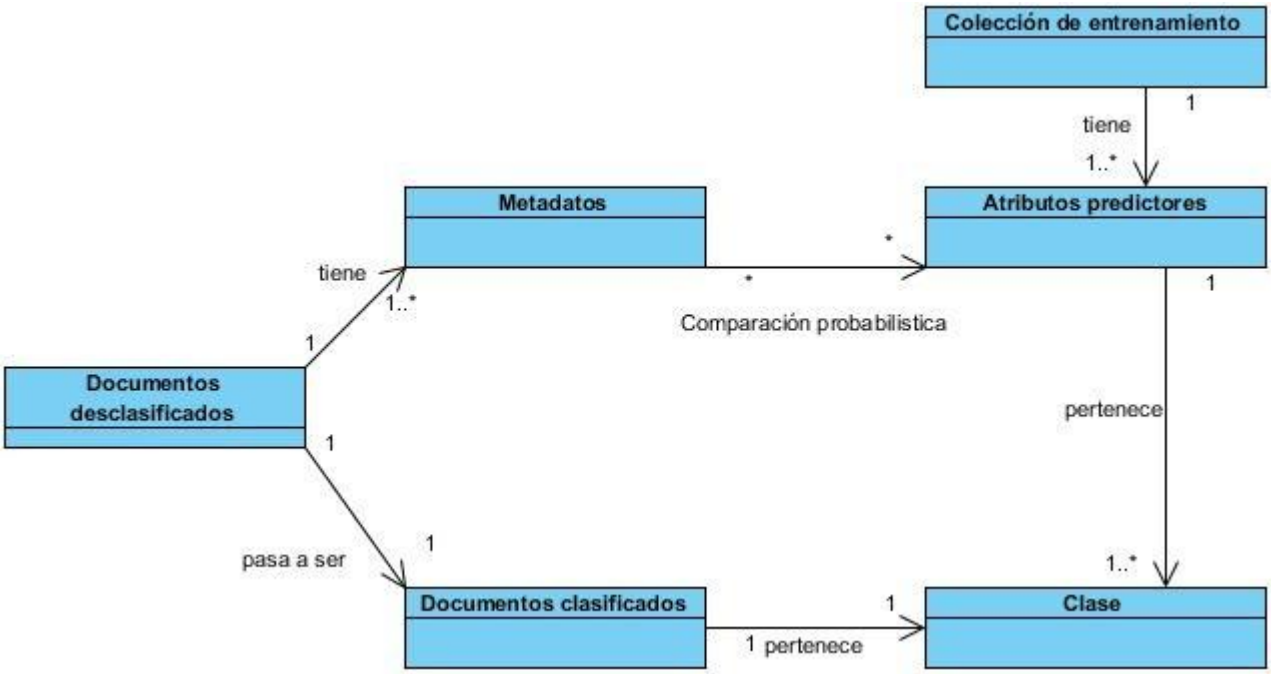

Figura 1: Modelo de dominio.

2.5.1 Glosario de Términos del Dominio

Metadatos: Información que es almacenada dentro de la base de datos con una referencia al documento completo.

Documentos desclasificados: Documentos que están en la base de datos pero que aún no han sido clasificados.

Documentos clasificados: Documentos a los cuales se le ha aplicado el algoritmo y calculado la probabilidad de pertenencia a las clases.

Colección de entrenamiento: Grupo de documentos previamente clasificados. Se emplean para determinar la probabilidad de que un nuevo documento pertenezca a las clases donde están clasificados los documentos de esta colección de entrenamiento.

Atributos predictores: Términos significativos que se extraen de los documentos pertenecientes a la colección de entrenamiento. Se utilizan para el cálculo de probabilidades.

Clase: Grupo al cual pertenece un documento en relación a su clasificación.

2.6 Requerimientos

Los requerimientos para la realización de la propuesta de solución se exponen a continuación.

Requisitos funcionales:

Un requisito funcional define el comportamiento interno del software: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas.

- RF1: Clasificar un documento.
  - ✓ RF1.1: Seleccionar atributos del documento.
  - ✓ RF1.2: Calcular probabilidades.
  - ✓ RF1.3: Asignar documento a la clase que pertenece.

Requisitos no funcionales:

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos, ya que éstos corresponden a los requisitos funcionales. Por tanto, se refieren a todos los requisitos que ni describen información a guardar, ni funciones a realizar (VIDAL, 2011).

Legales:

- Las herramientas seleccionadas para el desarrollo del producto están respaldadas por licencias libres, bajo las condiciones de software libre. Para la herramienta Visual Paradigm la cual no es libre se utiliza la licencia que posee la universidad.

- La aplicación y toda la documentación generada pertenecen al grupo de proyecto Biblioteca del Centro de Informatización Universitaria y a la Universidad de las Ciencias Informáticas.

Hardware:

Procesamiento:

- Para su correcto funcionamiento se requiere de un CPU Intel Pentium o compatible.

Memoria RAM:

- Para su correcto funcionamiento se requiere de 2 Gb mínimos

Capacidad en disco:

- Para su correcto funcionamiento se requiere de 80 Gb disponibles.

Soporte:

- Actualizar la colección de entrenamiento periódicamente o cuando se haga necesario para mejorar el rendimiento.

Restricciones de diseño e implementación:

- El sistema se desarrollará utilizando como lenguaje de programación PHP v5.3.8.
- El entorno de desarrollo integrado será Netbeans v7.2.
- Para la modelación del sistema se utilizará Visual Paradigm v8.0.
- Metodología de desarrollo de software XP, usando el lenguaje de modelación UMLv2.1.

2.7 Historias de Usuario

Son la técnica utilizada para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer (CANÓS *et al.*, 2003).

Historia de Usuario	
Código:HU01	**Nombre Historia de Usuario:** Clasificar documento
Modificación de Historia de Usuario Número: Ninguna	
Referencia: RF1	
Programador: Alex Rodríguez Serrano	**Iteración Asignada:** 1
Prioridad: Alta	**Puntos Estimados:** 12 días
Riesgo en Desarrollo: Alto	**Puntos Reales:** 11 días
Descripción: La Historia de Usuario inicia cuando el usuario accede al sistema, para clasificar los documentos. El usuario introduce los datos que necesita el clasificador. El sistema clasifica los documentos en la clase correspondiente.	
Observaciones:	
Prototipo interfaz:	

Tabla 1: Historia de Usuario clasificar documento.

Historia de Usuario	
Código:HU02	**Nombre Historia de Usuario:** Seleccionar atributos del documento
Modificación de Historia de Usuario Número: Ninguna	
Referencia: RF1.1	
Programador: Alex Rodríguez Serrano	**Iteración Asignada:** 1
Prioridad: Media	**Puntos Estimados:** 4 días
Riesgo en Desarrollo: Media	**Puntos Reales:** 4 días
Descripción: La Historia de Usuario inicia cuando el sistema comienza el proceso de clasificar los documentos. El sistema obtiene los datos relevantes eliminando los datos que son poco significativos para el clasificador.	
Observaciones:	
Prototipo interfaz:	

Tabla 2: Historia de Usuario seleccionar atributos del documento.

Historia de Usuario	
Código:HU03	**Nombre Historia de Usuario:** Calcular probabilidades
Modificación de Historia de Usuario Número: Ninguna	
Referencia: RF1.2	
Programador: Alex Rodríguez Serrano	**Iteración Asignada:** 1
Prioridad: Alta	**Puntos Estimados:** 5 días
Riesgo en Desarrollo: Alta	**Puntos Reales:** 4 días
Descripción: La Historia de Usuario inicia cuando el sistema termina la selección de los atributos significativos del documento. El sistema realiza el cálculo de probabilidades	
Observaciones:	
Prototipo interfaz:	

Tabla 3: Historia de Usuario calcular probabilidades.

Historia de Usuario	
Código:HU04	**Nombre Historia de Usuario:** Asignar documento a la clase que pertenece.
Modificación de Historia de Usuario Número: Ninguna	
Referencia: RF1.3	
Programador: Alex Rodríguez Serrano	**Iteración Asignada:** 1
Prioridad: Alta	**Puntos Estimados:** 2 día
Riesgo en Desarrollo: Alta	**Puntos Reales:** 2 días
Descripción: La Historia de Usuario inicia cuando el sistema culmina el cálculo de probabilidades. El sistema selecciona la clase que mayor probabilidad alcanza. El sistema asigna el documento a la clase seleccionada.	
Observaciones:	
Prototipo interfaz:	

Tabla 4: Historia de Usuario asignar documento a la clase que pertenece.

2.8 Estándares de codificación

Los estándares de codificación se utilizan generalmente para una mejor organización y productividad en la programación de proyectos en equipos o en solitario. Para el desarrollo del clasificador de documentos propuesto, se emplearán algunos de una serie de estándares establecidos por los desarrolladores de Zend Framework[2] (SITIO OFICIAL DE ZEND FRAMEWORK). Estos estándares seleccionados se adaptan al lenguaje de programación con el que se trabajará.

2.8.1 Convenciones de nombres

Clases

- Los nombres de las clases deben comenzar con mayúscula.
- Los espacios en los nombres de las clases no están permitidos, cuando el nombre contenga más de una palabra se debe delimitar con un guion bajo.

Archivos

- Para los nombres de archivos, sólo están permitidos caracteres alfanuméricos, guiones bajos y el carácter guión ("-"). Los espacios están estrictamente prohibidos.
- Cualquier archivo que contiene código PHP debe terminar con la extensión ".Php".

Funciones y métodos

- Los nombres de funciones sólo pueden contener caracteres alfanuméricos. Los números están permitidos en los nombres de las funciones, pero no se aconseja en la mayoría de los casos.
- Los nombres de funciones deben empezar siempre con una letra minúscula.
- Cuando los nombres de funciones contienen más de una palabra se deben delimitar con guion bajo. Los espacios están estrictamente prohibidos.

[2] Es un *framework* de código abierto para desarrollar aplicaciones web y servicios web con PHP 5

Variables

- Los nombres de variables pueden contener caracteres alfanuméricos. Los números están permitidos en los nombres de las variables pero no se aconseja en la mayoría de los casos.
- Los nombres de variables deben empezar siempre con una letra minúscula

2.8.2 Estilos de código

Demarcación del código

- El código PHP debe estar siempre limitado por las etiquetas estándar de PHP:

```
1. <? Php
2.
3. ?>
```

Cadena literal

- Cuando una cadena es literal (no contiene sustitución de variables), las 'comillas simples' siempre deben ser usadas para delimitar la cadena:

```
$a = 'Example String';
```

Cadenas literales que contienen apóstrofes

- Cuando una misma cadena contiene apóstrofos, es permitido delimitar la cadena con "comillas dobles". Esto es especialmente útil para las sentencias SQL:

```
$sql = "SELECT `id`, `name` from `people` "
    . "WHERE `name`='Fred' OR `name`='Susan'";
```

Sustitución de variables

- La sustitución de variables es permitida en cualquiera de estas formas:

```
$greeting = "Hello $name, welcome back!";
```

$greeting = "Hello {$name}, welcome back!";

Concatenación de cadenas

- Las cadenas deben ser concatenadas usando el operador "." . Un espacio debe añadirse siempre antes y después de este operador:

  $ Compañía = 'Zend. ". 'Tecnologías';

2.8.3 Clases

Declaración de clases

- Las clases deben ser nombradas de acuerdo a la convención de nombres de Zend Framework.Sólo una clase está permitida en cada archivo PHP.

Variables miembro de clases

- Las variables miembro de las clases deben ser nombradas de acuerdo a las convenciones de denominación de variables de Zend Framework.

2.8.4 Funciones y métodos

Declaración de funciones y métodos

- Las funciones deben ser nombradas de acuerdo con las convenciones de nombres de función de Zend Framework.
- No se permite el espacio entre el nombre de función y el paréntesis de apertura para los argumentos.

Uso de funciones y métodos

- Los argumentos de la función deben estar separados por un único espacio final después del delimitador de coma.

  Tres_argumentos (1, 2, 3);

Comentarios en las funciones.

- Todas las funciones deben tener un comentario, antes de su declaración, explicando que hacen. Ningún programador debería tener que analizar el código de una función para conocer su utilidad. Tanto el nombre como el comentario que acompañe a la función deben bastar para ello.

Precedencia de operadores.

- Esta regla es de gran ayuda para entender el código de la aplicación. Se recomienda siempre usar paréntesis para organizar las operaciones, principalmente en las matemáticas. Ejemplo:

  //Si no se tienen muchos conocimientos matemáticos puede ser complicado entender esta operación.
  $\$bool = (\$i < 7 \;\&\&\; \$j > 8 \;||\; \$k == 4);$
  //En cambio, si se programa así, es más sencillo.
  $\$bool = (\$i < 7 \;\&\&\; (\$j > 8 \;||\; \$k == 4));$

2.8.5 Sentencias de control

If / else / Elseif

- Dentro de las sentencias condicionales entre los paréntesis, los operadores deben estar separados por espacios para facilitar la lectura.

2.9 Diagrama de clases

Para lograr una implementación más fácil de la propuesta se opta por seguir el paradigma de la Programación Orientada a Objetos (POO). La problemática es modelada mediante clases con sus respectivos atributos y las relaciones existentes entre ellas. A continuación se expone el diagrama de clases perteneciente a la solución.

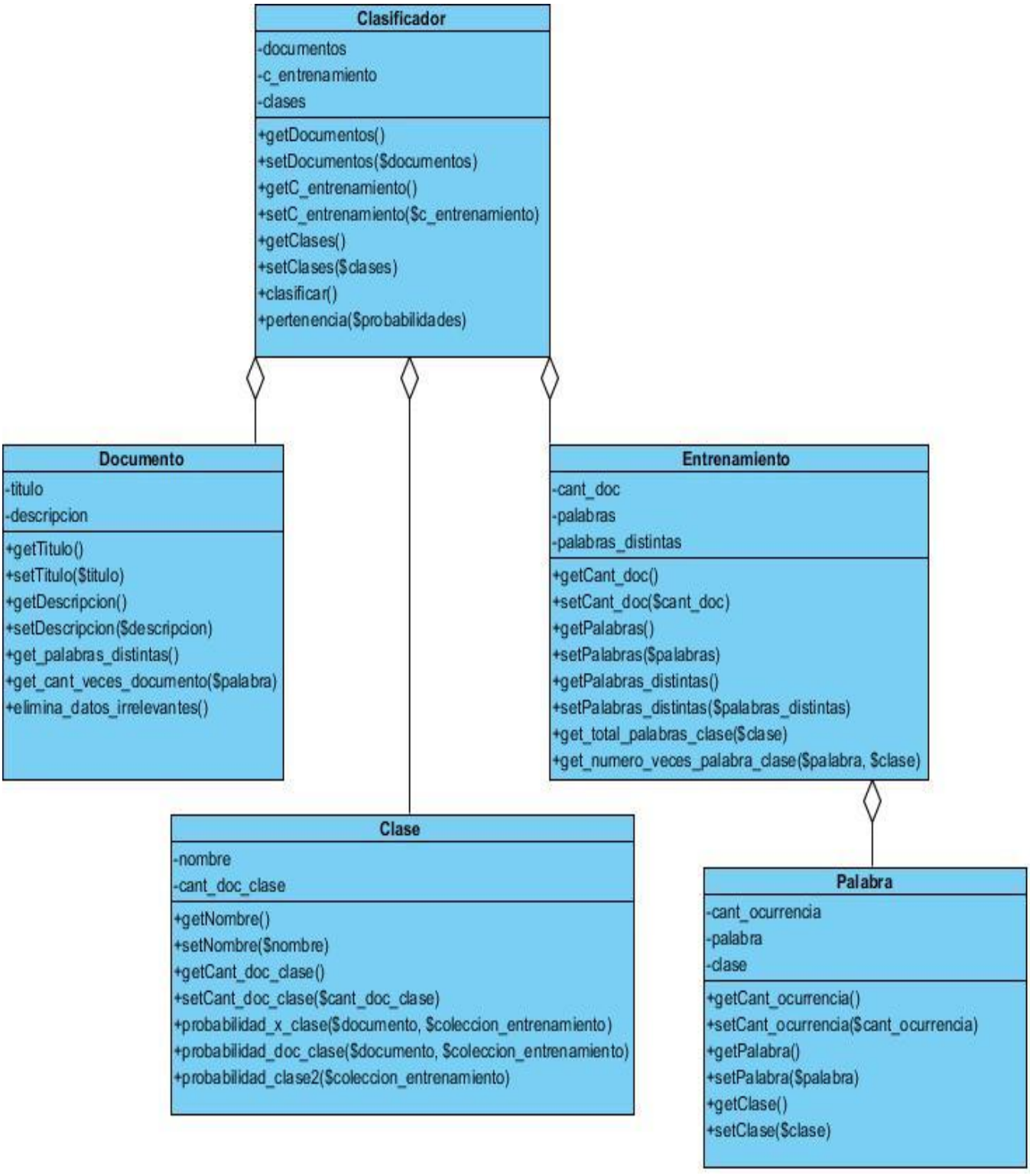

Figura 2: Diagrama de clases.

2.10 Conclusiones parciales

Luego de analizar el modelo Naive Bayes se logra comprender la base matemática del mismo con el fin hacer factibles las tareas de implementación. Al analizar y diseñar la solución se puede concluir que las fases de la implementación a las que más tiempo se les debe dedicar por sobresaliente importancia son a la del cálculo de probabilidades y a la de seleccionar la clase a la cual pertenece el documento. La selección de los estándares de codificación logra que se obtenga una mejor organización del código, además de facilitar la comprensión para futuras mejoras realizadas por otros programadores.

Capítulo 3 Implementación y pruebas del clasificador de documentos mediante el modelo Naive Bayes

3.1 Introducción

En el presente capítulo se expondrán los elementos fundamentales para la implementación del algoritmo propuesto, se detallará el código de la solución para cada una de las funciones. También se describen las pruebas a realizar al algoritmo implementado, definiéndose el tipo de pruebas, técnicas y métodos a utilizar.

3.2 Descripción de las fases principales del clasificador

A continuación se realiza una descripción detallada de los elementos y pasos fundamentales para el desarrollo del algoritmo.

3.2.1. Colección de entrenamiento.

Luego de analizar los clasificadores supervisados de documentos, se puede constatar que su principal componente es la colección de entrenamiento. De la exactitud con que esté creada, dependerán en gran medida los resultados del clasificador.

Para el desarrollo del clasificador se utiliza la colección de documentos 20 Newsgroups. Esta es una colección de aproximadamente 20.000 documentos particionados bien uniformes a través de 20 grupos de noticias, cada uno correspondiente a un tema diferente. Según Gabrilovich y Markovitch (2004) fue recopilada originalmente por Ken Lang en el año 1995.

La colección se encuentra dentro de los conjuntos de documentos disponibles en forma libre y gratuita más populares para clasificación (APARICIO y ACUÑA, 2009).

Algunos de los grupos de noticias están muy estrechamente relacionados entre sí (por ejemplo comp.sys.ibm.pc.hardware / comp.sys.mac.hardware). En la siguiente tabla se muestra una lista de los 20 grupos de noticias, agrupados por materias.

Comp.graphics Comp.os.ms-windows.misc Comp.sys.ibm.pc.hardware Comp.sys.mac.hardware Comp.windows.x	Rec.autos Rec.motorcycles Rec.sport.baseball Rec.sport.hockey	Sci.crypt Sci.electronics Sci.med Sci.space
Misc.forsale	Talk.politics.misc Talk.politics.guns Talk.politics.mideast	Talk.religion.misc Alt.atheism Soc.religion.christian

Tabla 5: Grupos de la colección de documentos.

Partiendo de que varios de los grupos están estrechamente relacionados se decide escoger solamente los grupos siguientes:

Grupo	Categoría
comp.grapics	Gráficos de computadora
rec.sport.baseball	Beisbol
sci.electronics	Electrónica
talk.politics.guns	Política de armas

Tabla 6: Grupos seleccionados para la colección de entrenamiento.

3.2.2. Selección de atributos significativos.

Los clasificadores suelen degradar su comportamiento ante atributos irrelevantes o redundantes; este problema está asociado a lo que se conoce como la maldición de la dimensionalidad que involucra no solo los problemas de los algoritmos para enfrentar grandes estructuras de datos, sino también cambios en la estructura de los datos en sí mismos (BENJAMINI y LESHNO, 2005).

La selección de atributos, selección de variables o reducción de atributos es una técnica comúnmente utilizada en aprendizaje computacional que involucra la selección de un subconjunto de los atributos más relevantes para mejorar el rendimiento del clasificador.

Existen dos grupos de metodologías que se utilizan para la selección de atributos: los métodos de filtro (SAEYS *et al.*, 2007) y los métodos de envoltura(KOHAVI y JOHN, 1997).

Los métodos de envoltura tienden a generar mejores subconjuntos de atributos, pero el costo computacional hace muy difícil su aplicación para conjunto de datos muy grandes, también presentan problemas de sobreajuste(REUNANEN *et al.*, 2003).

Los métodos de filtro, por otro lado, analizan las características de los datos sin considerar el algoritmo de clasificación (SÁNCHEZ *et al.*, 2007). Entre las ventajas de la selección por filtro se puede mencionar(SÁNCHEZ *et al.*, 2007):Mejorar el comportamiento del algoritmo de aprendizaje.

- Facilitar la comprensión de los datos.
- Disminución de los requerimientos de memoria y uso de la CPU al entrenar el algoritmo de aprendizaje.

Debido a estas características se implementó un método de filtro para la selección de atributos del algoritmo clasificador propuesto.

Esta fase se debe realizar tanto a la colección de entrenamiento en el proceso de aprendizaje, como a los documentos que se van a clasificar. Esto posibilita que el clasificador no se detenga a analizar palabras que no brindan información para determinar a cual clase se debe enviar el documento en cuestión.

3.2.3. Cálculo de probabilidades.

Se debe analizar en profundidad todas las ecuaciones matemáticas referentes a probabilidades a priori y posteriori, relacionadas con el modelo Naive Bayes que se introducen en el capítulo 2. Luego de este análisis se procede a programar las funciones que computen estas ecuaciones matemáticas. Este constituye un paso de suma significación para que el clasificador brinde resultados cercanos a la realidad.

3.2.4. Entrenamiento del modelo.

Para lograr resultados relevantes en el proceso de clasificación, una vez programadas las funciones que realicen los cálculos matemáticos, se debe realizar la etapa de entrenamiento. Para esta etapa se seleccionó el 70% de los documentos de cada clase debido a que las prestaciones computacionales con las que se realizaron las pruebas no son las idóneas para realizar estas tareas. Los resultados de esta fase se reflejan en la siguiente tabla.

	GC	B	E	PG	TOTAL
GC	86.67(612)	0.00(0)	0.00(0)	13.33(95)	707
B	0.00(0)	84.15(589)	3.55(25)	12.30(86)	700
E	12.41(87)	0.00(0)	77.33(546)	10.26(72)	705
PG	1.88(14)	0.00(0)	0.00(0)	98.12(722)	736
					2848

Tabla 7: Resultados de la etapa de entrenamiento.

Como se puede observar, en esta fase, los textos Gráficos de Computadora (GC) son clasificados correctamente en un 86,67% de los casos. Los textos de Beisbol (B) son clasificados correctamente en un 84,15%. Para el caso de Electrónica (E), el método en esta fase alcanza una clasificación correspondiente al 77,33% de los textos. Por último, para Política de Armas (PG), el método presenta un resultado que alcanza al 98,12% de los textos. Luego de concluida

la fase se calcula el porcentaje de clasificación correcta (PCC) para esta etapa considerando el número de aciertos dividido por el total de textos de la fase:

$$PCC = \frac{612 + 589 + 546 + 722}{2848} = 86{,}69\%$$

3.3 Análisis del código de la solución.

A continuación se presenta el código de los principales métodos de la propuesta de solución. Además, para facilitar su comprensión, se argumenta cual es la funcionalidad que realizan. Solo se hace énfasis en los métodos que se vinculan al algoritmo presentado, pero todos los métodos son de suma importancia para el funcionamiento del clasificador.

3.3.1 Método para selección de atributos.

Este método realiza una de las tareas más importantes, ya que su propósito es eliminar los signos de puntuación y las palabras que no son relevantes en un texto, pero sí muy frecuentes. A estas se les llama palabras vacías (*stop words*) y generalmente son las preposiciones, conjunciones, disyunciones y los términos que son muy comunes.

Para que se obtengan buenos resultados en esta fase, es necesario comprender que un mismo término puede ser más o menos significativo en un contexto que en otro, de manera que tendrá diferente peso en un documento que en otro. Adicionalmente, términos que aparecen en casi todos los documentos parecen poco significativos para recuperar documentos a partir de ellos. También influyen el tamaño o número de términos de cada documento. No parece lo mismo que un término aparezca cinco veces en un documento largo, de muchas páginas; a que aparezca también cinco veces en un documento corto, de dos páginas.

Para consultar todas las palabras que fueron seleccionadas como irrelevantes para el proceso de clasificación, puede remitirse a la sección de Anexos.

```php
function elimina_datos_irrelevantes() {
    $verbos = array(Verbos que no aportan información);
    $filtro = array(Palabras vacias);
    $signos1 = array(',', "'", '/', '.', '\\', ';', ':', '~', '$', '%', '#', '`', '*', '>', '<', '^');
    $signos = array(',', '.', ';', ':', '_');
    $espacios = array('      ', '     ', '    ', '   ', '  ', ')', '(', '[', ']', '{', '}', '"', '¿', '?', '!', '¡');
    $espacios1 = array('_', '-', '&', '=', '+', '@', '|');

    $cadena = trim($this->descripcion);
    $cadena = preg_replace('/\s(?=\s)/', '', $cadena);
    $cadena = preg_replace('/[\n\r\t]/', ' ', $cadena);
    $cadena = preg_replace("/[\n|\r|\n\r]+/", "", $cadena);
    $string = strtolower($cadena);
    $string = str_replace($espacios, ' ', $string);
    $string = str_replace($signos, '', $string);
    $string = preg_replace("/[0-9]/", "", $string);
    $string = preg_replace("/[a-z]+@[a-z]+/", '', $string);
    $string = preg_replace("/@[a-z]+/", '', $string);
    $string = str_replace($signos1, '', $string);
    $string = str_replace($espacios1, ' ', $string);
    $string = preg_replace('/\b[a-zA-Z]{1,3}\b/', '', $string);
    $palabras_array = explode(' ', $string);
    $resultado = array_diff($palabras_array, $filtro);
    $palabras = array_diff($resultado, $verbos);
    $tabla = array();
    foreach ($palabras as $palabra) {
        if (!in_array($palabra, $tabla)) {
            $numero = substr_count($string, ' ' . $palabra . ' ');
            if ($numero >= 3) {
                for ($i = 0; $i < $numero; $i++) {
                    array_push($tabla, $palabra);
                }
            }
        }
    }
    $salida = implode(' ', $tabla);
    return $salida;
}
```

Figura 3: Eliminar datos irrelevantes.

3.3.2 Método para calcular la probabilidad de pertenencia del documento en una clase

Es una función que, conociendo la clase, calcula la probabilidad de pertenencia del documento que se está clasificando en una clase conocida.

Para ello se necesita obtener varios valores de la colección de entrenamiento, tales como:

- N_{ik} es el número de veces que el término aparece en los documentos de la clase.
- N_i es el número total de palabras en los documentos de la clase.
- M es el tamaño del vocabulario (número de palabras distintas en la colección de entrenamiento).

También es necesario obtener los datos relacionados al documento en cuestión:

- n_{jk} es el número de veces que la palabra aparece en el documento.

```
function probabilidad_doc_clase($documento, $coleccion_entrenamiento) {
  $multiplicatoria = 1.0;
  $palabras = $documento->get_palabras_distintas();
  $ni = $coleccion_entrenamiento->get_total_palabras_clase($this->getNombre());
      foreach ($palabras as $palabra) {
      $nik = $coleccion_entrenamiento->get_numero_veces_palabra_clase($palabra, $this->getNombre());
      $ptkci = ($nik + 1) / ($ni + $coleccion_entrenamiento->getPalabras_distintas());
      $njk = $documento->get_cant_veces_documento($palabra);
      $multiplicatoria *= pow($ptkci, $njk);
      }
 return $multiplicatoria;
}
```

Figura 4: Calcular la probabilidad de que un documento pertenezca a una clase.

3.3.3 Método para calcular la probabilidad a priori de la clase

La probabilidad de una clase se puede calcular dividiendo la cantidad de documentos que son asignados a dicha clase en la colección de entrenamiento entre el total de documentos que existen en la colección de entrenamiento.

Para calcular la probabilidad a priori de la clase se necesita obtener los siguientes valores:

- $N_{i,doc}$ es el número de documentos en la colección de entrenamiento que se le asigna a la clase c_i.
- N_{doc} es el número de documentos en la colección de entrenamiento.

```
function probabilidad_clase($coleccion_entrenamiento) {
  $ndoc = $coleccion_entrenamiento->getCant_doc();
  $n = $this->cant_doc_clase;
 return $n / $ndoc;
}
```

Figura 5: Calcular la probabilidad a priori de la clase.

3.4 Pruebas

Las pruebas pueden considerarse como un conjunto de actividades en las cuales un sistema o componente es ejecutado bajo condiciones o requerimientos especificados.

Para validar el algoritmo implementado se realizaron 2 tipos de pruebas, las pruebas de caja blanca para verificar el funcionamiento del código y las pruebas de precisión para comprobar la efectividad de algoritmo.

3.4.1 Pruebas de caja blanca

La prueba de caja blanca, denominada a veces prueba de caja de cristal, es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba (PRESSMAN, 2002).

Con estas pruebas se pueden comprobar los caminos lógicos, bucles y condiciones examinando así el estado del programa para informar de la situación real de la calidad del software. En esta investigación se realizaron las comprobaciones para casos críticos como el cálculo de probabilidades y selección de la clase a la que pertenece el documento.

Camino básico

El método de prueba de camino básico se suele aplicar a un código del sistema o software que se esté desarrollando**.**

En esta investigación se decide realizar este método a la función encargada de calcular la probabilidad de pertenencia del documento en la clase que se está analizando y a la función encargada de seleccionar la probabilidad máxima para la clasificación del documento.

Caso de prueba 1: Calcular probabilidad de pertenencia de un documento en una clase.

Paso 1: Dibujar el grafo de flujo asociado a partir del código fuente.

```
function probabilidad_doc_clase($documento, $coleccion_entrenamiento) {
  $multiplicatoria = 1.0;
  $palabras = $documento->get_palabras_distintas();
  $ni = $coleccion_entrenamiento->get_total_palabras_clase($this->getNombre());
      foreach ($palabras as $palabra) {
      $nik = $coleccion_entrenamiento->get_numero_veces_palabra_clase($palabra, $this->getNombre());
      $ptkci = ($nik + 1) / ($ni + $coleccion_entrenamiento->getPalabras_distintas());
      $njk = $documento->get_cant_veces_documento($palabra);
      $multiplicatoria *= pow($ptkci, $njk);
      }
 return $multiplicatoria;
}
```

Figura 6: Código del caso de prueba 1. Calcular probabilidad de pertenencia de un documento en una clase.

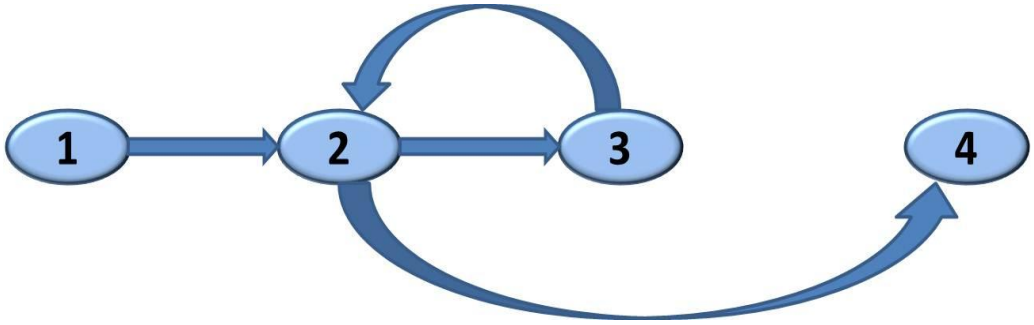

Figura 7: Grafo del caso de prueba 1. Calcular probabilidad de pertenencia de un documento en una clase.

Paso 2: Cálculo de complejidad ciclomática.

V (G)= Número de Aristas – Número de Nodos + 2. V (G) = 4 – 4 + 2 = 2

Paso 3: Caminos básicos.

Camino básico 1: 1-2-3-4

Paso 4: Caso de prueba para el camino 1: 1-2-3-4

Entrada: Documento

Colección de entrenamiento

Resultado esperado: El sistema calcula la probabilidad de pertenencia del documento en la clase que se está analizando.

Caso de prueba 2: Seleccionar probabilidad máxima.

Paso 1: Dibujar el grafo de flujo asociado a partir del código fuente.

```
function pertenencia($probabilidades) {
    $clase = array();
    foreach ($probabilidades as $documento => $probabilidad) {
        $maximo = max($probabilidad);
        foreach ($probabilidad as $valor => $value) {
            if ($value == $maximo) {
                $clase["$documento"] = $valor;
            }
        }
    }
    return $clase;
}
```

Figura 8: Código del caso de prueba 2. Seleccionar máxima probabilidad.

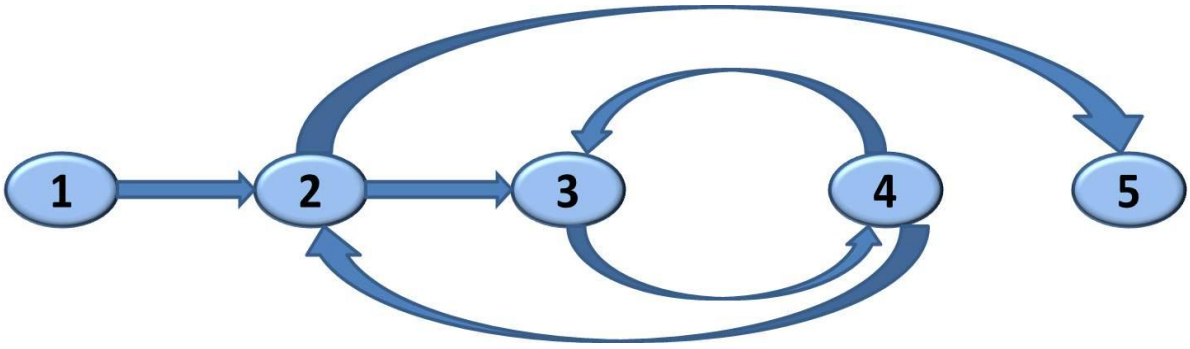

Figura 9: Grafo del caso de prueba 2. Seleccionar máxima probabilidad.

Paso 2: Cálculo de complejidad ciclomática.

V (G)= Número de Aristas – Número de Nodos + 2. V (G) = 6 – 5 + 2 = 2

Paso 3: Caminos básicos.

Camino básico 1: 1-2-3-4-5

Paso 4: Caso de prueba para el camino 1: 1-2-3-4-5
Entrada: Arreglo con las probabilidades de pertenencia de un documento a cada clase
Resultado esperado: El sistema selecciona la máxima probabilidad.

3.4.2 Pruebas de precisión

Esta prueba está enfocada en medir la precisión teniendo en cuenta los resultados que el sistema emite. Es exactamente la precisión la variable que se definió para medir los resultados de la investigación, por lo cual de esta prueba derivan los principales resultados de la misma.

Para realizar estas pruebas al algoritmo, se almacenaron los datos que necesita el clasificador en una Base de Datos. Para ello se utiliza el Gestor de Base de Datos MySQL en su versión 5.5.29 y el servidor web Apache en su versión 2.2. Los datos se almacenaron como se explica a continuación.

La Base de Datos utilizada para probar el algoritmo cuenta con 4 tablas donde se guarda toda la información de los documentos de la colección de entrenamiento y de los documentos que se necesitan clasificar.

En la tabla **clase** se encuentran las clases en las cuales están clasificados los documentos de la colección de entrenamiento, así como la cantidad de documentos de la colección de entrenamiento asociados a cada clase. Los nuevos documentos se clasifican en alguna de las clases de esta tabla. Se decidió añadir una clase llamada otras para en caso de que las probabilidades de pertenencia de un documento sean iguales en todas las clases poder clasificar el documento. A esta última clase no se le asocia ningún documento de la colección de entrenamiento.

En la tabla **ocurrencia** se encuentran los atributos significativos. Estos atributos son las palabras relevantes para la clasificación que se encuentran en los documentos de entrenamiento. Estos vienen acompañados de la cantidad de veces que se repiten en la clase a la que están asignadas.

En la tabla **documentos_desclasificados** se almacenan de forma provisional los datos de los documentos que aún no se han procesado. Estos documentos se insertan en la tabla

documentos_clasificados luego de que se les aplica el algoritmo y se determina cuál es la clase a la que se asocian. En esta nueva tabla se almacenarán los datos pertenecientes al documento y adicionalmente se representarán con el identificador de la clase en que fueron clasificados. A continuación se muestra la modelación de la estructura de datos.

Calculo de precisión

Para realizar el cálculo de la precisión se tomó la ecuación utilizada por Romero(2010):

$$P = \frac{DBC}{DC} \qquad (7)$$

Dónde:

- ***P*** es la precisión.
- ***DBC*** son los documentos bien clasificados.
- ***DC*** son todos los documentos clasificados.

Para realizar la prueba se parte de la clasificación de 10 documentos por cada una de las clases alcanzando un total de 40 documentos.

Para comprobar si la técnica escogida para disminuir la independencia condicional brinda buenos resultados, se realizan dos iteraciones de la prueba. En la primera iteración se clasifican los documentos sin aplicar dicha técnica. Los resultados de esta primera prueba de precisión se reflejan en la siguiente tabla.

Primera iteración

	GC	B	E	PG	TOTAL
GC	70.00%(7)	0.00%(0)	10.00%(1)	20.00(2)	10
B	0.00(0)	70.00(7)	0.00(0)	30.00(3)	10
E	00.00(0)	0.00(0)	70.00(7)	30.00(3)	10
PG	00.00(0)	0.00(0)	00.00(0)	100.00(10)	10
					40

Tabla 8: Resultados de las pruebas de precisión. 1ra iteración.

Como se puede observar la tabla 8, los textos de la categoría Gráficos de Computadora (GC) son clasificados correctamente en un 70,00% de los casos. Los textos de Beisbol (B) son clasificados correctamente en un 70,00%. Para el caso de Electrónica (E), el método en esta fase alcanza una clasificación correspondiente en el 70,00% de los textos. Por último, para Política de Armas (PG), el método presenta un resultado que alcanza al 100.00% de los textos. Luego de concluida la fase se calcula la precisión del algoritmo para esta iteración considerando el número de aciertos dividido por el total de textos de la fase:

$$P = \frac{DBC}{DC} = \frac{31}{40} = 0{,}775$$

Segunda iteración

Para la segunda iteración, realizada para comprobar si la técnica empleada para disminuir la hipótesis de independencia condicional es eficiente, se mantienen los mismos documentos empleados en la iteración anterior pero se comprueban los resultados aplicando dicha técnica.

	GC	B	E	PG	TOTAL
GC	40.00%(4)	10.00%(1)	30.00%(3)	10.00%(1)	10
B	00.00%(0)	30.00%(3)	70.00%(7)	00.00%(0)	10
E	10.00%(1)	00.00%(0)	90.00%(9)	00.00%(0)	10
PG	00.00%(0)	10.00%(1)	10.00%(1)	80.00%(8)	10
					40

Tabla 9: Resultados de las pruebas de precisión. 2da iteración.

Como se puede observar la tabla 9, los textos de la categoría Gráficos de Computadora (GC) son clasificados correctamente en un 40,00% de los casos. Los textos de Beisbol (B) son clasificados correctamente en un 30,00%. Para el caso de Electrónica (E), el método en esta fase alcanza una clasificación correspondiente en el 90,00% de los textos. Por último, para Política de Armas (PG), el método presenta un resultado que alcanza al 80.00% de los textos. Luego de concluida la iteración se calcula la precisión del algoritmo para esta segunda iteración considerando el número de aciertos dividido por el total de textos de la fase:

$$P = \frac{DBC}{DC} = \frac{24}{40} = 0{,}60$$

Con los resultados de esta prueba se evidencia una diferencia negativa de 17,5% con respecto a la primera iteración donde no se aplica el proceso de disminuir la independencia condicional. Debido a dicha diferencia negativa, se decide no incluir la técnica descrita por Qiang (2010).

3.5 Conclusiones parciales

Al término de las fases de implementación y prueba, se aprecia que dentro de las fases necesarias para el desarrollo de un clasificador supervisado, se destaca la creación de la colección de entrenamiento y con esta la selección de atributos. De lo representativos que puedan ser los datos asignados en la colección para cada clase depende la efectividad del algoritmo. En este capítulo, se puede apreciar detalladamente los principales métodos de la propuesta de solución. Esto posibilita que se entienda con claridad cuáles fueron los procedimientos seguidos en el proceso de implementación. También se puede constatar, basándose en las pruebas realizadas al código, que los caminos básicos del algoritmo se ejecutan al menos una vez en cada uno de los métodos analizados.

En cuanto a precisión luego de realizar las pruebas, se desecha la opción de aplicar la técnica seleccionada para intentar disminuir la independencia entre todas las palabras. Dicha técnica fue rechazada debido a que al ejecutar el algoritmo con la técnica seleccionada, se nota en el porcentaje de clasificación adecuada una diferencia negativa de 17.5% respecto a la iteración realizada sin aplicarle el suavizado.

Conclusiones

- Del estudio realizado sobre clasificación supervisada, se logró conocer sus principales características, los algoritmos utilizados para esta tarea, así como que pueden ser utilizados en diversas áreas.
- Con el análisis y diseño del algoritmo, se alcanzó un punto de partida para su implementación. Luego de culminar estas dos fases se alcanza una correcta organización para el desarrollo del algoritmo.
- Con la eliminación de palabras vacías y signos de puntuación, se logró reducir de una manera adecuada la cantidad de términos que se analizan en el cálculo de probabilidades.
- El almacenamiento de palabras significativas junto con su aparición en cada clase posibilitó confeccionar una colección de entrenamiento que proporciona mejores probabilidades y reducen significativamente la complejidad computacional.
- Con la implementación se logró desarrollar un algoritmo que permite clasificar los documentos en la clase a la cual pertenece.
- Las pruebas realizadas al código permitieron verificar que los caminos básicos de las funciones principales se ejecutaran al menos una vez y que brindaran el resultado esperado.
- En relación a las pruebas de precisión los resultados no fueron los esperados pues se tuvo que desechar la técnica de suavizado seleccionada para disminuir la hipótesis de independencia condicional asumida por Naive Bayes. Se decide no utilizar el método de suavizado puesto que al realizar pruebas sin aplicarlo, se obtuvieron mejores resultados que aplicándolo.
- Finalmente se cumplió el objetivo planteado de implementar un sistema de clasificación supervisada de documentos con el propósito de contribuir a mejorar el sistema de navegabilidad del repositorio documental de la UCI.

.

Recomendaciones

- Experimentar con otro mecanismo que trate adecuadamente el *ruido* en los documentos, como por ejemplo, palabras escritas incorrectamente (errores ortográficos).
- Experimentar con otras técnicas de suavizado para intentar reducir la hipótesis de independencia incondicional.
- Para futuras mejoras, crear la colección de entrenamiento solo con los términos relevantes para mejorar los tiempos de cómputo.
- Crear una colección de entrenamiento en español para que pueda ser usado en el repositorio documental de la UCI. En este tema debería existir el apoyo de un especialista en bibliotecología para lograr una colección donde cada clase esté realmente representada por los términos que más ocurren en esta y por ende mayor porciento de predicción poseen.

Referencias Bibliográficas

ÁLVAREZ, R. J. D. D. *Clasificación Automática de Textos usando reducción de clases basada en prototipos* Tesis de maestria, INAOEP, 2009. Disponible en: http://ccc.inaoep.mx/labtl/uploads/Main/TesisMaestria-JuanAlvarez.pdf. [Consultado: Febrero del 2013]

ANGUIANO, H. E. *Naive Bayes Multinomial para Clasificación de Texto Usando un Esquema de Pesado por Clases.* 2009. Disponible en: http://www.google.com.cu/url?sa=t&rct=j&q=anguiano+2009+Reuters-21578+&source=web&cd=1&cad=rja&ved=0CCsQFjAA&url=http%3A%2F%2Fccc.inaoep.mx%2F~esucar%2FClases-mgp%2FProyectos%2FMGP_RepProy_Abr_29.pdf&ei=To6rUYqWNI2Lswb07IHIBg&usg=AFQjCNHgyqb9Mjh0LQcc2MlmFEV7AqNWeA&bvm=bv.47244034,d.Yms. [Consultado: Enero del 2013].

APARICIO, C. R. K. y ACUÑA, F., EDGAR. *Clasificación Semi-Supervisada de Documentos* 2009 Disponible en: http://www.iiis.org/CDs2009/CD2009CSC/CISCI2009/PapersPdf/C758MD.pdf. [Consultado: Marzo del 2013].

ARMAÑANZAS, A. R. *Medidas de filtrado de seleccion de variables mediante la plataforma Elvira.* 2004. Disponible en: http://www.sc.ehu.es/ccwbayes/members/ruben/msth.pdf. [Consultado: Febrero del 2013]

BALBONTÍN, G. Á. y SÁNCHEZ, M. J. J. *SPNER – Reconocedor de entidades nombradas para el español.* 2004. Disponible en: http://www.esi.uem.es/~jmgomez/plenum/plenum4/04.pdf. [Consultado: Enero del 2013].

BECK, K. *Extreme programming.* 1999. Disponible en: http://laerer.rhs.dk/henrikh/SYME-2007f/xp.pdf. [Consultado: Marzo del 2013].

BENJAMINI, Y. y LESHNO, M. Statistical methods for data mining. *Data Mining and Knowledge Discovery Handbook.*, 2005, Disponible en: http://link.springer.com/chapter/10.1007/0-387-25465-X_25. ISSN: 1384-5810. [Consultado: Enero del 2013].

BERMEJO, S. *Learning with nearest neighbour classifiers.* Doctoral, Universitat Politécnica de Catalunya, 2000. Disponible en: http://www.tdx.cat/TDX-0408103-145004/. [Consultado: Diciembre del 2013]

CALABRIA, E. *Implementación de Redes bayesianasmediante una aplicación para ladetección y el filtrado de SPAM.* Facultad de Ingenieria. Universidad de Buenos Aires, 2006. Disponible en: http://es.scribd.com/doc/74134548/Deteccion-de-SPAM-mediante-redes-bayesianas. [Consultado: Enero del 2013]

CANÓS, J.; LETELIER, P., *et al. Metodologías Ágiles en el desarrollo de Software.* 2003. Valencia: Disponible en: http://noqualityinside.com.ar/nqi/nqifiles/XP_Agil.pdf. [Consultado: Febrero del 2013].

CASTELLANO, A. *Exactitud y presición.* 2009. Disponible en: http://www.upaep.cesat.com.mx/index.php?option=com_content&view=article&id=28:exactitud-y-precision&catid=11:metrologia&Itemid=14. [Consultado: Enero del 2013].

CEDEÑO, G. Y. Módulo de representación de información de navegabilidad en entornos virtuales. *Serie Científica*, 2010, vol. 3, nº. 2, Disponible en: http://publicaciones.uci.cu/index.php/SC/article/view/318. ISSN: 2306-2495. [Consultado: Junio del 2013].

CESTNIK, B.; KONONENKO, I., *et al.* A knowledge elicitation tool for sophisticated users. En *The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases - ECML 1987*. Disponible en: http://journalogy.net/Publication/510642/assistant-86-a-knowledge-elicitation-tool-for-sophisticated-users. [Consultado: Diciembre del 2012].

COHEN, K. B. y HUNTER, L. Getting started in text mining. *PLoS Computational Biology*, 2008, vol. 4, nº. 1, Disponible en: http://dx.plos.org/10.1371/journal.pcbi.0040020. ISSN: 1553-7358. [Consultado: Diciembre del 2012].

CHOW, C. y LIU, C. *Approximating discrete probability distributions with dependence trees.* 1968. Disponible en: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1054142. [Consultado: Diciembre del 2012].

DEFINICIÓN. *Definición de lenguaje de programación*. Disponible en: www.definicion.org/lenguaje-de-programacion. [Consultado: Enero del 2013].

DUDA, R. y HART, P. *Pattern classification and scene analysis.* 1973. Pattern classification and scene analysis. Disponible en: http://www.ica.luz.ve/~enava/redesn/ebooks/DHS/Versi%F3n%20PS/DHSChap5.ps. [Consultado: Noviembre del 2012]

ECHEVERRÍA, F.; CARRERA, M., *et al. Desarrollo de una Extranet para Importaciones.* 2012. Disponible en: http://repositorio.maeug.edu.ec/handle/123456789/30. [Consultado: Diciembre del 2012].

FIGUEROA, C.; ALONSO BERROCAL, J. L., *et al. ALGUNAS TECNICAS DE CLASIFICACION AUTOMATICA DE DOCUMENTOS.* 2005. Universidad de Salamanca: Grupo REINA. Disponible en: http://reina.usal.es/papers/figuerola2005algunas.pdf. [Consultado: Febrero del 2013]

FRESNO, F. V. *Representación Autocontenida de Documentos HTML: una propuesta basada en Combinaciones Heurísticas de Criterios. .* doctoral, Escuela Superior de Ciencias Experimentales y Tecnología, Departamento de Ingeniería Telemática y Tecnología

Electrónica,. Disponible en: . Universidad Rey Juan Carlos, 2006. Disponible en: http://www.escet.urjc.es/~vfresno/phd_sp.html. [Consultado: Febrero del 2013]

FRIEDMAN, N.; GEIGER, D.*, et al.* Bayesian network classifiers. *1573-0565*, 1997, vol. 29, nº. 2-3, Disponible en: http://link.springer.com/article/10.1023/A%3A1007465528199. [Consultado: Enero del 2013].

GABRILOVICH, E. y MARKOVITCH, S. Text categorization with many redundant features: Using aggressive feature selection to make SVMs competitive with C4. 5. En *2004*. Disponible en: http://dl.acm.org/citation.cfm?id=1015388. [Consultado: Febrero del 2013].

GIBERT, K.; SPATE, J.*, et al.* Data mining for environmental systems. . *decision support systems*, 2008 vol. 3, p. 205-228. Disponible en: http://www.athanasiadis.info/pdf/papers/emsds2008data.pdf. ISSN: 0167-9236. [Consultado: Febrero del 2013].

GIULIANELLI, D.; RODRÍGUEZ, R.*, et al. Situación Global de Gobernabilidad Electrónica en Sitios Web Municipales.* 2008. Disponible en: http://www.imaginar.org/ecollecter/fullpapers/p60-SituacionGlobalDeGobernabilidadElectronica.pdf.

GÓMEZ, V. F.; TORRES, G. M.*, et al.* Algoritmo de selección de atributos basado en utilidad incremental y análisis de redundancia: aplicación a datos biológicos. . *AVANCES EN INTELIGENCIA ARTIFICIAL*, 2011, vol. 1, isponible en: http://aepia.aic.uniovi.es/revista/index.php/aia/article/view/931/754. ISSN: 1988-3064. [Consultado: Diciembre del 2012].

GONZÁLEZ, Y. D. y ROMERO, Y. F. Patrón Modelo-Vista-Controlador. . *Revista Telem@tica*, 2012, vol. 11, Disponible en: http://revistatelematica.cujae.edu.cu/index.php/tele/article/view/15/10. ISSN: 1729-3804. [Consultado: Marzo del 2013].

GOUTTE, C. y GAUSSIER, E. A probabilistic interpretation of precision, recall and Fscore, with implication for evaluation. En *Advances in Information Retrieval, 27th European Conference on IR Research. New York. 2005*. Disponible en: http://link.springer.com/chapter/10.1007/978-3-540-31865-1_25. [Consultado: Junio del 2013].

HASSAN, M. Y. y MARTÍN, F. F. J. Sistemas de Clasificación de Información. *No Solo Usabilidad*, 2004, vol. 3, Disponible en: http://www.nosolousabilidad.com/articulos/sistemas_clasificacion.htm. ISSN: 1886-8592. [Consultado: Diciembre del 2013].

HE, J.; TAN, A. H.*, et al. On quantitative evaluation of clustering systems.* 2003. Disponible en: http://www.google.com/books?hl=es&lr=&id=WJsd7Mz7zJEC&oi=fnd&pg=PA105&dq=%22On+Quantitative+Evaluation+of+Clustering+Systems%22&ots=XrWXOGcYZd&sig=RWfLf6v-ggxZ6uPJB3ajL03un6g. [Consultado: Febrero del 2013].

HERNÁNDEZ, O. E. *El lenguaje unificado de modelado(UML).* 2008. Valencia: Disca. Disponible en: http://www.disca.upv.es/enheror/pdf/ActaUML.PDF. [Consultado: Enero del 2013]

JEFFRIES, R. E.; ANDERSON, A.*, et al. Extreme programming installed.* Addison-Wesley. 2001. Extreme programming installed. Disponible en: http://www.google.com/books?hl=es&lr=&id=5ZuPjdO8LLoC&oi=fnd&pg=PR13&dq=%22Extreme+Programming+Installed%22&ots=OHVzscS31A&sig=xQ5nbgPtIsVBeRO5GoDWtWxHxlc. [Consultado: Marzo del 2013]

KEOGH, E. J. y PAZZANI, M. Learning augmented Bayesian classifiers: a comparison of distribution-based and non distribution-based approaches. En *Proceedings of the seventh international workshop on artificial intelligence and statistics. 1999*. Disponible en: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.55.7726&rep=rep1&type=pdf. [Consultado: Marzo del 2013].

KOHAVI, R. y JOHN, G. Wrappers for Feature Subset Selection Artificial Intelligence. *Artificial intelligence*, Marzo del 2013 1997, vol. 97, nº. 1, Disponible en: http://www.sciencedirect.com/science/article/pii/S000437029700043X. ISSN: 1435-5655. [Consultado: Febrero del 2013].

KONONENKO, I. Semi-naive bayesian classifier. En Kodratoff, Y. (editor). *Machine Learning — EWSL-91.* Springer Berlin Heidelberg. 1991, vol. 482, p. 206-219. Disponible en: http://dx.doi.org/10.1007/BFb0017015. [Consultado: Febrero del 2013].

LARRANAGA, P.; INZA, I.*, et al. Tema 6. Clasificadores Bayesianos.* 2007. Universidad del País Vasco-Euskal Herriko Unibertsitatea: Disponible en: http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/t6bayesianos.pdf. [Consultado: Enero del 2013].

LASCANO, E.; ORDOÑEZ, L. F. R.*, et al. Desarrollo de un MASHUP comercial para la publicación y búsqueda de bienes inmuebles en quito integrando distintas APIS públicas y proveyendo de las interfaces necesarias para que sea consumido desde cualquier tipo de aplicación informática.* 2011. Disponible en: http://repositorio.espe.edu.ec/handle/21000/4723. [Consultado: Febrero del 2013]

LAURITZEN, S. L. y SPIEGELHALTER, D. J. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society Series B*, 1988, vol. 50, Disponible en: http://www.jstor.org/stable/10.2307/2345762. ISSN: 1467-9868. [Consultado: Enero del 2013].

MÉNDEZ, G. O. y AQUINO, R. Y. *PROPOSED USE OF NAÏVE BAYESIAN NETWORK FOR ANOMALIES DETECTION IN ETHERNET TRAFFIC.* 2011. Ciudad de la Habana: Disponible en: http://www.informaticahabana.cu/node/1219. [Consultado: Marzo del 2013].

MORENO, L. J. y VANEGAS, P. S. *Sistema de información para el registro de capacitaciones y pago de impuestos y tratamiento de extintores de la Corporación Pro Desarrollo de*

Girardot. 2009. Disponible en: http://hdl.handle.net/10656/1307. [Consultado: Febrero del 2013].

MURILLO, A. F. *Herramientas Case* 2008. Disponible en: http://www.inei.gob.pe/biblioineipub/bancopub/Inf/Lib5103/Libro.pdf. [Consultado: Febrero del 2013].

ORNELLA, L. A. *Códigos Correctores de Error en Problemas de Clasificación Multiclase de Datos de Marcadores Moleculares.* doctoral, Universidad Nacional de Rosario, 2010. Disponible en: http://sedici.unlp.edu.ar/handle/10915/19907. [Consultado: Marzo del 2013]

ORTEGA, M. I. Precisión y Exáctitud. En *Taller de calidad de datos en Bases de datos de Biodiversidad. España. 2008*. Disponible en: www.gbif.es/ficheros/Precision_y_exactitud.ppt. [Consultado: Junio del 2013].

PAZZANI, M. *Constructive induction of cartesian product attributes.* 1998. Disponible en: http://link.springer.com/chapter/10.1007/978-1-4615-5725-8_21. [Consultado: Enero del 2013].

PAZZANI, M. *Searching for dependencies in Bayesian classifiers.* 1996. Disponible en: http://link.springer.com/chapter/10.1007/978-1-4612-2404-4_23. [Consultado: Enero del 2013].

PRESSMAN, R. S. *Ingeniería de Software. Un enfoque práctico.* 2002. Ingeniería de Software. Un enfoque práctico. [Consultado: Enero del 2013]

QIANG, G. An Effective Algorithm for Improving the Performance of Naive Bayes for Text Classification. En *Conferencia Internacional sobre Investigación y Desarrollo de la Informática. 2010*. p. 699-701. Disponible en: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5489530. [Consultado: Enero del 2013].

RAE. *Precisión.* 2001. Disponible en: http://lema.rae.es/drae/?val=precision. [Consultado: Junio del 2013].

RAMÍREZ, C. A. *Exactitud y precisión*. Puebla, Mexico: Disponible en: http://www.upaep.cesat.com.mx/index.php?option=com_content&view=article&id=28:exactitud-y-precision&catid=11:metrologia&Itemid=14. [Consultado: Junio del 2013].

REUNANEN, J.; GUYON, I.*, et al.* Overtting in Making Comparisons Between Variable Selection Methods. . *Journal of Machine Learning Research*, 2003, vol. 3, p. 1371-1382. Disponible en: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.11.1942. ISSN: 1533-7928. [Consultado: Enero del 2013].

RIPOLL, M. D. A. y PÉREZ, C. Y. *Asignacion automatizada de categorias tematicas al contenido textual de documentos HTML.* Trabajo de Diploma, Universidad de las Ciencias Informáticas, 2008. Disponible en:

http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_1258_08. [Consultado: Diciembre del 2012]

ROMERO, L. A. E. *Document classification models based on bayesian networks.* Tutor: De Campos, I. L. M. y Fernandez, L. J. M. Departament of computer science and artificial intelligence. University of Granada, 2010. Disponible en: http://decsai.ugr.es/~aeromero/files/thesis.pdf. [Consultado: Enero del 2013]

SAEYS, Y.; INZA, I., *et al.* A review of feature selection techniques in bioinformatics. . *Bioinformatics*, 2007, p. 2507-2517. Disponible en: http://bioinformatics.oxfordjournals.org/content/23/19/2507.short. ISSN: 1460-2059. [Consultado: Enero del 2013].

SAHAMI, M. Learning limited dependence Bayesian classifiers. En *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. 1996*. Disponible en: http://www.bibsonomy.org/bibtex/2b17115f59af78099e951efcc395e55c8/dblp. [Consultado: Abril del 2013].

SALINAS, C. P. y HISTCHFELD, N. *Lenguaje unificado de modelado*. Disponible en: http://www.dcc.uchile.cl/~psalinas/uml/introduccion.html. [Consultado: Enero del 2013].

SÁNCHEZ, M., N.; ALONSO, B., A. *, et al.* Filter Methods for Feature Selection. *Springer Berlin*, 2007, p. 178-187. Disponible en: http://link.springer.com/chapter/10.1007/978-3-540-77226-2_19. ISSN: 1432-0584. [Consultado: Marzo del 2013].

SEBASTIANI, F. Machine Learning in Automated Text Categorization. . *ACM Computing Surveys (CSUR)*, 2002, vol. 34, nº. 1, Disponible en: http://www.isti.cnr.it/People/F.Sebastiani/Publications/ACMCS02.pdf ISSN: 1557-7341 [Consultado: Marzo del 2013].

SITIO OFICIAL DE NETBEANS. Disponible en: http://netbeans.org/features/index.html. [Consultado: Noviembre del 2012].

SITIO OFICIAL DE ZEND FRAMEWORK. *Zend Framework Coding Standard for PHP*. Disponible en: http://framework.zend.com/manual/1.12/en/coding-standard.html. [Consultado: Marzo del 2013].

SOUMEN, C. *Mining the Web: discovering knowledge from hypertext data.* San Francisco, CA: Morgan Kaufmann. 2003. Mining the Web: discovering knowledge from hypertext data. ISBN 1558607544.

STEPHENS, M. y ROSENBERG, D. Extreme programming refactored: the case against XP. *Computing Reviews*, 2004, vol. 45, nº. 6, p. 330. Disponible en: http://www.ulb.tu-darmstadt.de/tocs/113946430.pdf. ISSN: 1530-6585. [Consultado: Marzo del 2013].

SUÁREZ, C. A. Resolución de la Ambigüedad Semántica de las palabras mediante Modelos de Probabilidad de Máxima Entropía. *Procesamiento de lenguaje natural*, 2005, vol. 34,

Disponible en: http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/viewFile/3047/1540. ISSN: 1989-7553. [Consultado: Enero del 2013].

VIDAL, L. M. *Sistema de Comunicación y Difusión del Observatorio de Derechos y Centro de Recursos de Información. .* 2011. Disponible en: http://upcommons.upc.edu/handle/2099.1/12491. [Consultado: Marzo del 2013].

VILLA, B. H. y ALFONSO, S. I. R. Biblioteca híbrida: el bibliotecario en medio del tránsito de lo tradicional a lo moderno. *ACIMED*, 2005, vol. 13, Disponible en: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352005000200005&lng=es&nrm=iso. ISSN: 1561-2880. [Consultado: Febrero del 2013].

VOUTSSÁS, J.; RAMÍREZ, J. A.*, et al. Panel: Bibliotecas digitales.* 2002. p.11-25. Disponible en: http://www.google.com/books?hl=es&lr=&id=jEZfmTKEv0EC&oi=fnd&pg=PA11&dq=%22bibliotecas+digitales%22+Juan+Voutssas+jose+antonio&ots=W9k8YxFLbA&sig=ap6AqD6fzc-ZFaJ029I0lelcZZc. [Consultado: Mayo del 2013].

WIS. *Usage and Population Statistics*. 2012 Disponible en: http://www.internetworldstats.com/stats.htm. [Consultado: Junio del 2013].

WITTEN, I. H. y EIBE, F. E. *Data Mining. Practical machine learning tools and techniques 2nd* 2005. Disponible en: http://www.google.com/books?hl=es&lr=&id=QTnOcZJzlUoC&oi=fnd&pg=PR17&dq=+IH+Witten,+E+Frank,+Data+Mining:+Practical+Machine+Learning+Tools+and+Techniques+(Morgan+Kaufmann,+San+Francisco,+2005)&ots=3gnwarVhP9&sig=YOVli7K30rGMnZHnXV_YwY0e-qs. [Consultado: Diciembre del 2012].

YANG, Y. An evaluation of statistical approaches to text categorization. *Information retrieval*, 1999, vol. 1, nº. 1-2, p. 69-90. Disponible en: http://link.springer.com/article/10.1023/A%3A1009982220290. ISSN: 1573-7659. [Consultado: Enero del 2013].

YANG, Y.; ADELSTEIN, S. J.*, et al.* Target discovery from data mining approaches. . *Drug Discovery today*, 2009, vol. 14, p. 147-154. Disponible en: http://www.far.fiocruz.br/farmanguinhos/images/stories/mestrado/2011/Target_discovery_from_data_mining.pdf. ISSN: 1359-6446. [Consultado: Diciembre del 2012].

Bibliografía consultada

ÁLVAREZ, J. D. D. Clasificación Automática de Textos usando reducción de clases basada en prototipos Tesis de maestria, INAOEP, 2009. Disponible en: http://ccc.inaoep.mx/labtl/uploads/Main/TesisMaestria-JuanAlvarez.pdf. [Consultado: Febrero del 2013]

ANGUIANO, E. Naive Bayes Multinomial para Clasificación de Texto Usando un Esquema de Pesado por Clases. 2009. Disponible en: http://www.google.com.cu/url?sa=t&rct=j&q=anguiano+2009+Reuters-21578+&source=web&cd=1&cad=rja&ved=0CCsQFjAA&url=http%3A%2F%2Fccc.inaoep.mx%2F~esucar%2FClases-mgp%2FProyectos%2FMGP_RepProy_Abr_29.pdf&ei=To6rUYqWNI2Lswb07IHIBg&usg=AFQjCNHgyqb9Mjh0LQcc2MlmFEV7AqNWeA&bvm=bv.47244034,d.Yms. [Consultado: Enero del 2013].

APARICIO, C. R. K. y ACUÑA, F., EDGAR. Clasificación Semi-Supervisada de Documentos. 2009. Disponible en: http://www.iiis.org/CDs2009/CD2009CSC/CISCI2009/PapersPdf/C758MD.pdf. [Consultado: Marzo del 2013].

ARMAÑANZAS, A. R. Medidas de filtrado de seleccion de variables mediante la plataforma Elvira. 2004. Disponible en: http://www.sc.ehu.es/ccwbayes/members/ruben/msth.pdf. [Consultado: Febrero del 2013]

BALBONTÍN, G. Á. y SÁNCHEZ, M. J. J. SPNER – Reconocedor de entidades nombradas para el español. 2004. Disponible en: http://www.esi.uem.es/~jmgomez/plenum/plenum4/04.pdf. [Consultado: Enero del 2013].

BENJAMINI, Y. y LESHNO, M. Statistical methods for data mining. Data Mining and Knowledge Discovery Handbook., 2005, Disponible en: http://link.springer.com/chapter/10.1007/0-387-25465-X_25. ISSN: 1384-5810. [Consultado: Enero del 2013].

BERMEJO, S. Learning with nearest neighbour classifiers. Doctoral, Universitat Politécnica de Catalunya, 2000. Disponible en: http://www.tdx.cat/TDX-0408103-145004/. [Consultado: Diciembre del 2013]

CABRERA, F. A. y COUTIN, D. A. Las bibliotecas digitales: Parte I. Consideraciones teóricas. ACIMED, 2005, Disponible en: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352005000200004&lng=es. ISSN: 1561-2880. [Consultado: Noviembre del 2012].

CALABRIA, E. El clasificador Naive Bayes un ejemplo práctico. 2006, Disponible en: http://es.scribd.com/doc/74134548/13/El-clasificador-Naive-Bayes-un-ejemplo-practico. [Consultado: Diciembre del 2012].

CARLOS G. FIGUEROLA; RAQUEL GÓMEZ, et al. Spanish Monolingual Track: The Impact of Stemming on Retrieval. Computer Science, 2002, vol. 2406, Disponible en: http://link.springer.com/chapter/10.1007/3-540-45691-0_23. ISSN: 0302-9743. [Consultado: Enero del 2013].

CASTELLANO, A. Exactitud y presición. 2009. Disponible en: http://www.upaep.cesat.com.mx/index.php?option=com_content&view=article&id=28:exactitud-y-precision&catid=11:metrologia&Itemid=14. [Consultado: Enero del 2013].

COHEN, K. B. y HUNTER, L. Getting started in text mining. PLoS Computational Biology, 2008, vol. 4, nº 1, Disponible en: http://dx.plos.org/10.1371/journal.pcbi.0040020. ISSN: 1553-7358. [Consultado: Diciembre del 2012].

CHOW, C. y LIU, C. Approximating discrete probability distributions with dependence trees. 1968. Disponible en: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1054142. [Consultado: Diciembre del 2012].

DUDA, R. y HART, P. Pattern classification and scene analysis. 1973 Disponible en: http://www.ica.luz.ve/~enava/redesn/ebooks/DHS/Versi%F3n%20PS/DHSChap5.ps. [Consultado: Noviembre del 2012]

FIGUEROA, C.; ALONSO BERROCAL, J. L., et al. ALGUNAS TECNICAS DE CLASIFICACION AUTOMATICA DE DOCUMENTOS. 2005. Universidad de Salamanca: Grupo REINA. Disponible en: http://reina.usal.es/papers/figuerola2005algunas.pdf. [Consultado: Diciembre del 2012]

FRIEDMAN, N.; GEIGER, D., et al. Bayesian network classifiers., 1997, vol. 29, nº 2, Disponible en: http://link.springer.com/article/10.1023/A%3A1007465528199. ISSN 1573-0565, [Consultado: Enero del 2013].

GÓMEZ, V. F.; TORRES, G. M., et al. Algoritmo de selección de atributos basado en utilidad incremental y análisis de redundancia: aplicación a datos biológicos. . AVANCES EN INTELIGENCIA ARTIFICIAL, 2011, vol. 1, Disponible en: http://aepia.aic.uniovi.es/revista/index.php/aia/article/view/931/754. ISSN: 1988-3064. [Consultado: Diciembre del 2012].

GOUTTE, C. y GAUSSIER, E. A probabilistic interpretation of precision, recall and Fscore, with implication for evaluation. En Advances in Information Retrieval, 27th European Conference on IR Research. New York. 2005. Disponible en: http://link.springer.com/chapter/10.1007/978-3-540-31865-1_25. ISSN: 1573-0565. [Consultado: Junio del 2013].

HASSAN, M. Y. y MARTÍN, F. F. J. Sistemas de Clasificación de Información. No Solo Usabilidad, 2004, vol. 3, Disponible en: http://www.nosolousabilidad.com/articulos/sistemas_clasificacion.htm. ISSN: 1886-8592. [Consultado: Diciembre del 2013].

KEOGH, E. J. y PAZZANI, M. Learning augmented Bayesian classifiers: a comparison of distribution-based and non distribution-based approaches. En Proceedings of the seventh international workshop on artificial intelligence and statistics. 1999. Disponible en: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.55.7726&rep=rep1&type=pdf. [Consultado: Marzo del 2013].

KONONENKO, I. Semi-naive bayesian classifier. En Kodratoff, Y. (editor). Machine Learning - EWSL-91. Springer Berlin Heidelberg. 1991, vol. 482, p. 206-219. Disponible en: http://dx.doi.org/10.1007/BFb0017015. [Consultado: Febrero del 2013].

PAZZANI, M. Constructive induction of cartesian product attributes. . 1998. Disponible en: http://link.springer.com/chapter/10.1007/978-1-4615-5725-8_21. ISSN: 1573-0565. [Consultado: Enero del 2013].

PAZZANI, M. Searching for dependencies in Bayesian classifiers. 1996. Disponible en: http://link.springer.com/chapter/10.1007/978-1-4612-2404-4_23. ISSN: 1573-0565. [Consultado: Enero del 2013].

PRESSMAN, R. S. Ingeniería de Software. Un enfoque práctico. 2002. Ingeniería de Software. Un enfoque práctico. [Consultado: Enero del 2013]

QIANG, G. An Effective Algorithm for Improving the Performance of Naive Bayes for Text Classification. En Conferencia Internacional sobre Investigación y Desarrollo de la Informática. 2010. p. 699-701. Disponible en: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5489530. [Consultado: Enero del 2013].

RENÉ, V. Clasificación de textos académicos en función de su contenido léxico-semántico. Signos, 2007, vol. 40, nº 63, Disponible en: http://www.scielo.cl/scielo.php?pid=S0718-09342007000100012&script=sci_arttext. ISSN: 0718-0934 [Consultado: Abril del 2013].

REUNANEN, J.; GUYON, I., et al. Overtting in Making Comparisons Between Variable Selection Methods. . Journal of Machine Learning Research, 2003, vol. 3, p. 1371-1382. Disponible en: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.11.1942. ISSN: 1533-7928. [Consultado: Enero del 2013].

RIPOLL, M. D. A. y PÉREZ, C. Y. Asignacion automatizada de categorias tematicas al contenido textual de documentos HTML. Trabajo de Diploma, Universidad de las Ciencias Informáticas, 2008. Disponible en: http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_1258_08. [Consultado: Diciembre del 2012]

ROMERO, L. A. E. Document classification models based on bayesian networks. Tutor: De Campos, I. L. M. y Fernandez, L. J. M. Departament of computer science and artificial intelligence. University of Granada, 2010. Disponible en: http://decsai.ugr.es/~aeromero/files/thesis.pdf. [Consultado: Enero del 2013]

SAHAMI, M. Learning limited dependence Bayesian classifiers. En Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. 1996. Disponible en: http://www.bibsonomy.org/bibtex/2b17115f59af78099e951efcc395e55c8/dblp. [Consultado: Abril del 2013].

SÁNCHEZ, M., N.; ALONSO, B., A. , et al. Filter Methods for Feature Selection. Springer Berlin, 2007, p. 178-187. Disponible en: http://link.springer.com/chapter/10.1007/978-3-540-77226-2_19. ISSN: 1432-0584. [Consultado: Marzo del 2013].

SEBASTIANI, F. Machine Learning in Automated Text Categorization. ACM Computing Surveys (CSUR), 2002, vol. 34, nº 1, Disponible en: http://www.isti.cnr.it/People/F.Sebastiani/Publications/ACMCS02.pdf ISSN: 1557-7341 [Consultado: Marzo del 2013].

YANG, Y. An evaluation of statistical approaches to text categorization. Information retrieval, 1999, vol. 1, nº 1-2, p. 69-90. Disponible en: http://link.springer.com/article/10.1023/A%3A1009982220290. ISSN: 1573-7659. [Consultado: Enero del 2013].

YANG, Y.; ADELSTEIN, S. J., et al. Target discovery from data mining approaches. . Drug Discovery today, 2009, vol. 14, p. 147-154. Disponible en: http://www.far.fiocruz.br/farmanguinhos/images/stories/mestrado/2011/Target_discovery_from_data_mining.pdf. ISSN: 1359-6446. [Consultado: Diciembre del 2012].

Glosario de términos

Colección de entrenamiento: Conjunto de documentos previamente clasificados. Se utilizan para predecir la probabilidad de pertenencia de un documento en alguna de las clases en las que están clasificados los documentos de esta colección.

Independencia Condicional: Se refiere a la independencia entre palabras asumidas por el algoritmo Naive Bayes.

Internet: Red de ordenadores a nivel mundial. Ofrece distintos servicios, como el envío y recepción de correo electrónico, la posibilidad de ver información en las páginas web, de participar en foros de discusión, de enviar y recibir ficheros, de charlar en tiempo real, entre otros.

Palabras vacías: Palabras que no aportan información para el proceso de clasificación. Entre estas suelen estar los artículos, pronombres, preposiciones, etc.

Repositorio: Sitio donde se almacena información en formato digital.

Anexos

tuesday	december	able	about	november	wednesday	october
friday	saturday	across	act	above	abst	thursday
according	accordingly	affects	after	actually	added	accordance
affected	affecting	almost	alone	afterwards	again	ad
ah	all	am	among	along	already	against
although	always	any	anybody	amongst	an	also
announce	another	anyways	anywhere	anyhow	anymore	and
anything	anyway	arise	around	apparently	approximately	anyone
at	aren't	available	away	as	aside	are
because	becomes	becoming	before	awfully	b	asking
began	begun	been	bear	bore	borne	became
begins	behind	being	below	beforehand	beginning	back
can	change	believe	bring	burn	buy	call
beyond	com	both	brief	beside	besides	beginnings
came	did	bought	catch	caught	chose	chosen
writes	interested	called	best	post	well	internet
c	d	cannot	cause	briefly	but	between
continue	cook	choose	clean	close	come	compare
co	due	comes	contain	causes	certain	by
decide	describe	cost	count	cry	cut	dance
couldn't	enough	date	different	containing	contains	certainly
become	begin	definitely	answer	arrive	ask	be
eat	end	destroy	die	do	drink	drive
eaten	feed	done	dream	dreamed	dreamt	ate
who's	didn't	don't	doesn't	widely	willing	wish
effect	body	during	e	does	doing	could
allow	opening	easily	true	false	sale	contact
ending	far	eight	eighty	each	ed	down
every	followed	especially	et	either	else	elsewhere
f	four	everyone	everything	etc	even	ever
find	finish	explain	fall	feel	fight	fill
gotten	gave	fed	felt	found	get	got
fix	gives	few	follows	every	ex	except
forth	having	following	further	fifth	first	five
have	hear	forget	forgive	give	go	hate

getting	hereby	from	goes	for	former	formerly
list	group	full	free	cant	based	note
kept	knew	given	went	gone	had	keep
has	hi	giving	him	further	g	gets
hereafter	how	he	however	h	happens	hardly
improve	jump	help	hide	hold	hope	hurt
hes	if	herein	immediate	hence	her	here
they've	do's	he's	let's	world	would	www
home	inc	hid	index	hereupon	hers	herself
in	ward	how	it	himself	his	hither
invention	just	im	last	hundred	i	id
j	l	indeed	lest	immediately	importance	important
knows	least	is	looking	information	instead	into
wouldn't	can't	it'll	that's	you	zero	your
you'll	won't	it'll	shouldn't	z	you'd	i'm
article	will	june	july	january	february	number
latterly	line	k	maybe	keeps	its	itself
listen	live	know	laugh	learn	leave	lie
made	said	known	learned	learnt	left	let
likely	makes	largely	mg	lately	kg	km
mainly	meanwhile	less	most	lets	later	latter
me	more	little	my	looks	like	liked
move	need	look	lose	love	make	meet
ml	mug	many	never	meantime	ltd	m
much	namely	merely	nobody	might	mean	means
name	needs	moreover	obtain	mostly	million	miss
myself	ninety	must	ok	necessary	mr	mrs
nine	nor	nay	ones	near	n	necessarily
now	here	neither	otherwise	never	nearly	next
off	often	no	over	non	new	noned
on	once	normally	part	not	none	nothing
ourselves	other	o	please	obtained	noted	of
prefer	prepare	occur	offer	open	pay	play
p	out	oh	provides	okay	obviously	omitted
per	page	one	ran	only	old	or
possibly	perhaps	others	recently	ought	onto	ours

probably	potentially	outside	relatively	overall	our	own
quite	promptly	pages	s	particular	owing	past
readily	less	placed	seem	plus	particularly	possible
read	receive	press	promise	pull	push	put
regardles	really	proud	several	present	poorly	primarily
resulting	regards	r	shows	q	previously	quickly
sec	results	recent	slightly	rather	refs	re
run	say	recognize	remember	repeat	rest	return
self	section	related	something	ref	respectively	regarding
similarly	selves	right	specifically	research	saying	resulted
sent	set	saw	seen	seek	sought	sold
sing	sit	see	sell	send	shout	show
somehow	should	seeing	sub	same	seeming	says
still	since	seven	sufficiently	seemed	she	seems
slain	slept	showed	shown	shut	slay	slew
t	someone	six	thanks	shall	significantly	shed
study	suggest	sleep	smile	speak	start	stay
the	soon	sorry	them	significant	some	similar
told	wrote	spoke	spoken	tear	tore	torn
excellent	sales	starter	fast	moving	entire	holds
there	stop	strongly	thereupon	so	sometimes	somebody
april	may	sunday	monday	august	september	march
think	throw	support	take	talk	teach	tell
to	sup	sure	thou	sometime	specify	somewhat
tries	than	thank	thru	specified	successfully	specifying
un	their	theirs	took	substantially	taking	such
upon	thereafter	thereby	unless	taken	that	tends
uses	this	thereto	used	themselves	then	that
going	newsgroup	thing	current	high	better	good
via	through	those	v	therefore	therein	thence
check	long	three	great	include	address	original
way	together	too	welcome	these	they	thereof
use	visit	touch	travel	treat	try	understand
when	truly	trying	where	though	towards	thousand
with	under	us	wherever	thus	two	tip
x	ups	usually	why	toward	unto	tried

problem	people	viewer	objects	simple	university	mail
was	were	want	wash	win	work	write
it's	using	whenever	words	twice	usefully	u
there's	we	who	yet	unlike	various	up
hasn't	whence	whose	yourselves	useful	w	very
i'll	whither	without	you're	value	whereas	wants
time	years	written	email	advance	started	carry
pretty	view	year	seconds	things	news	subject
haven't	y	yes	isn't	vs	which	while
you've	yours	yourself	what's	what	whole	whom

Anexo 1. Palabras irrelevantes para el proceso de clasificación.