UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL

DISEÑO Y DESARROLLO DE UN MÓDULO DE CLASIFICACIÓN DE PÁGINAS WEB EN BASE A LAS CARACTERÍSTICAS DE SU CONTENIDO UTILIZANDO TÉCNICAS DE MINERÍA DE DATOS

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL INDUSTRIAL

GONZALO ALEJANDRO FALLOUX COSTA

PROFESOR GUÍA:
JUAN DOMINGO VELÁSQUEZ SILVA

MIEMBROS DE LA COMISIÓN:
FELIPE ESTEBAN VILDOSO CASTILLO
ROCÍO BELÉN RUIZ MORENO
IGNACIO CALISTO LEIVA

SANTIAGO DE CHILE
2016

RESUMEN DE LA MEMORIA PARA OPTAR AL
TITULO DE: Ingeniero Civil Industrial
POR: Gonzalo Alejandro Falloux Costa
FECHA: 05/12/2016
PROFESOR GUIA: Juan Domingo Velásquez Silva

DESARROLLO DE MÓDULO PARA CLASIFICAR PÁGINAS WEB BASADO EN CONTENIDO

Este trabajo de título tiene por objetivo principal diseñar y desarrollar un módulo de clasificación de páginas web en base a las características de su contenido utilizando técnicas de minería de datos, lo que se traduce en la utilización de contenido HTML, análisis de texto visible de la página web y la incorporación de una variable que refleja la seguridad web según SSL como variables predictivas para la clasificación de páginas web.

El trabajo se realiza enmarcado en el proyecto AKORI del Web Intelligence Centre de la Facultad de Ciencias Matemáticas de la Universidad de Chile, el cual pretende desarrollar una plataforma computacional para mejorar el diseño y contenido de sitios web mediante el estudio de variables fisiológicas y la aplicación de minería de datos. La plataforma consiste en la implementación de un modelo que sea capaz de predecir mapas tanto de fijación ocular como de dilatación pupilar de manera rápida y precisa.

En esta etapa del proyecto AKORI es necesario mejorar el desempeño de las predicciones descritas, las cuales son realizadas en sitios web reales, de diseño y contenido muy variado. Además el comportamiento que se desea predecir es sobre usuarios de los que se desconoce su motivación para la navegación, lo cual a su vez altera tanto el comportamiento ocular como sus patrones de navegación.

Dado lo anterior se propone como hipótesis de investigación: *Es posible clasificar páginas web en base a las características de su contenido* para solucionar dos problemas fundamentales, por un lado la clasificación agrupa páginas web maximizando la varianza de páginas web entre clases y minimizando la varianza intra clase, lo cual debiese mejorar considerablemente el desempeño del modelo, puesto que predecir dentro de una clase en la cual los ejemplos tienen mayor similitud disminuye el rango de error, disminuyendo, a su vez el error estándar en la predicción. Por otro lado entrega información sobre la motivación del usuario en la web si se conoce el servicio que ofrece la página web, lo que si bien no es información completa para describir el comportamiento del usuario, puede ser una importante variable de apoyo.

Para el desarrollo del modelo se utiliza un juego de datos de 138 páginas web, escogidas según tráfico de usuarios Chilenos y luego se implementan cinco algoritmos de minería de datos para clasificar entre siete clases de páginas web. El algoritmo Naive Bayes obtiene el mejor desempeño, logrando un *accuracy* de 78.67 %, lo que permite validar la hipótesis de investigación.

Finalmente se concluye que se cumplen todos los resultados esperados y la hipótesis de investigación con resultados satisfactorios considerando la investigación actual.

A mis padres Luis y Patricia.

Agradecimientos

Finalizada esta etapa me gustaría agradecer no solo a todos los que estuvieron conmigo en el transcurso de este trabajo, si no que a cada persona que compartió conmigo la vida universitaria que de cierta forma me ayudó a llegar a este momento.

A mi familia que si bien no somos muchos siempre me hicieron sentir el mayor apoyo.

A mis amigos por lo grandes e innumerables momentos.

Al profesor Juan Velásquez por la oportunidad de trabajar con un gran equipo y de enfrentar desafíos fuera de la comodidad.

A todos los miembros y ex miembros del centro de investigación que siempre estuvieron dispuestos a ayudar.

Finalmente a todo aquel que me apoyó, en especial en los momentos más difíciles y que sigan recorriendo junto a mi este largo camino.

Tabla de Contenido

Capítulo 1

Introducción

La web afecta nuestras vidas de tal manera, que en la mayoría de los casos es difícil imaginar un futuro donde esta pierda importancia. La web revolucionó tan radicalmente el mundo de las comunicaciones llegando a un punto en que la humanidad se adapta al funcionamiento de la web y nos introduce a una realidad que es capaz de actuar como canal social, económico e incluso satisfacer necesidades espirituales.

La web (World Wide Web) es introducida públicamente en 1993, una fecha no muy lejana si se considera que actualmente[1] existen 3,631,124,813 usuarios[1], que corresponde al 49.5 % de la población mundial y alcanza el 79.9 % en el caso particular de Chile[2], lo que es aún más sorprendente es el hecho de que la web ha crecido a una tasa de 905 % en el caso mundial y 702 % en el caso de Chile desde el año 2000[2]. Es por esta razón que los estudios en las áreas de desempeño web y características propias de sitios web han surgido como temas relevantes en la investigación actual[3][4]. Es más, para organizaciones que utilizan internet para llevar a cabo tanto transacciones comerciales como no comerciales, su sitio web es su vitrina hacia el mundo, lo que inspira a su vez el estudio del diseño apropiado de características web y su influencia en el desempeño del sitio web[5].

Actualmente la web indexada contiene al menos 4.83 billones de páginas web[6], lo que hace necesario el desarrollo de técnicas para el manejo y recuperación de información relativo al contenido web, una solución a este problema es la clasificación web que además se encarga de problemas como el crawling focalizado, desarrollo de directorios web, análisis de *weblinks* para marketing contextualizado y análisis de tópicos estructurales en la web[7].

1.1. Antecedentes

Este trabajo se enmarca en el proyecto AKORI (Advanced Kernel for Ocular Research and web Intelligence),el cual es un proyecto financiado por INNOVA CORFO y desarrollado por Web In-

[1] Actualizado a 30 de Junio de 2016

telligence Centre (WIC[2]) de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile.

El proyecto AKORI busca generar un servicio del más alto nivel centrándose en el apoyo en la toma de decisiones con respecto a contenido y diseño de un sitio web, para esto se han realizado múltiples investigaciones partiendo por el gestor del centro de investigación WIC y quien comienza la línea de investigación de AKORI en el año 2004, Juan D. Velásquez. En su trabajo diseña una metodología que permite identificar palabras claves (*Website Keywords*) en un sitio web[8] que se definen como *"palabras que caracterizan el contenido de una página o sitio web dado"*.

Posteriormente, este trabajo es mejorado y no se limita sólo a texto, si no que considera contenido multimedia[9] en los denominados Web Objects: *çonjunto de palabras estructuradas o un recurso multimedia presente dentro de una página que posee metadatos que describan su contenido"*. En este momento nace una nueva línea de investigación que considera información empírica del usuario con el objetivo de aumentar la precisión de las metodologías anteriores. Para esto se considera por primera vez el eye-traker para medir el tiempo de atención en los objetos web[10].

El proyecto AKORI comienza con la investigación basada en comportamiento ocular y la web, puesto que se pretende realizar una plataforma capaz de predecir el comportamiento ocular humano en la web, esto motiva a más investigaciones en la línea de eye-tracking, pero esta vez considerando dilatación pupilar como variable predictiva[11][12], concluyendo que por sí sola esta variable no tiene mucha relevancia, puesto que es necesario conocer la relación del estímulo con la emocionalidad, lo cual es finalmente investigado por Slanzi[13], quien mediante estudios que consideran actividad eléctrica cerebral relaciona la emocionalidad con la dilatación pupilar, logrando de esta manera que esta variable si sea predictiva.

1.1.1. WIC

El Web Intelligence Centre es un centro de investigación aplicada que se enorgullece por ser capaz de proveer un servicio profesional, excelente y rápido para distintos rubros.

El WIC no solo realiza investigación aplicada, si no que sus estudios también son publicados en las principales revistas científicas tanto nacionales como internacionales y es parte de conferencias a nivel mundial relacionadas con Web Intelligence y Data Science.

Un pilar fundamental del WIC es promover el aprendizaje, es por eso que el WIC dicta cursos de orientación práctica acerca de las Tecnologías de Información y su aplicación en los negocios.

El WIC es miembro del Web Intelligence Consortium[3], el que agrupa distintos centros de investigación y desarrollo en Inteligencia Web de todo el mundo.

En su página web el WIC declara su misión, su visión y sus objetivos:

[2] `http://wic.uchile.cl`

[3] `http://wi-consortium.org/`, 14 de Abril del 2016

- ***Misión:*** Desarrollar investigación de frontera en el campo de Tecnologías de Información creando nuevas soluciones para abordar problemas complejos de ingeniería utilizando herramientas basadas en la Web de las Cosas.
- ***Visión:*** Ser un líder a nivel internacional en la investigación de tecnologías de información y comunicaciones aplicadas a la resolución de problemas del mundo real.
- Objetivos:
  - Publicar en las principales revistas, conferencias y editoriales relacionadas con Web Intelligence.
  - Proveer un servicio profesional, excelente y rápido para todos nuestros clientes.
  - Dictar cursos de orientación práctica acerca de las Tecnologías de Información y su aplicación en los negocios.

1.1.2. AKORI

AKORI es un proyecto que nace del trabajo conjunto entre investigadores del Departamento de Ingeniería Industrial de la Universidad de Chile y la Facultad de Medicina de la Universidad de Chile, que en conjunto desarrollan el FONDEF: "Plataforma informática basada en Web Intelligence y herramientas de análisis de exploración visual para la mejora de la estructura y contenido de sitios web".

AKORI es una plataforma enfocada en el análisis de las estructuras de las páginas web, entregando información acerca de la forma en que ciertos perfiles de usuarios, definidos por el cliente, visualizan dichas páginas. Lo anterior se ejecuta mediante la entrega de mapas de fijación ocular y dilatación pupilar, dando así información sobre la estructura que debería tener su página web para que tuviese mayor impacto en los visitantes.

1.1.2.1. Servicios de AKORI

AKORI pretende entregar diversos servicios para poder ayudar en la decisión del cómo se muestra la información en una página web. Estos servicios se encuentran en constante desarrollo e investigación, siendo algunos los siguientes:

1. ***Mapa de Fijación Ocular***

   Podemos construir el mapa de calor más probable de fijación ocular de tu sitio y puedes usarlo como un mapa de atención sobre el mismo para hacer los mejores cambios para mejorarlo. Las zonas rojas tienen gran probabilidad de ser vistas por los usuarios y las zonas azules tienen menos probabilidad. Zonas sin color tienen probabilidad nula de ser vistas.

2. ***Mapa de Dilatación Pupilar:***

   Un mapa de dilatación pupilar es una herramienta útil para tu sitio. Este mapa de calor puede ser un mapa de ayuda a la decisión, ya que es más probable obtener clicks en zonas con

mayor dilatación pupilar. Las zonas con alta probabilidad de ser clickeadas son rojas y las otras azules.

3. ***Mapa de Objetos Claves en un Sitio Web***

   El mapa de objetos claves en un sitio web muestra los objetos del DOM (algunos 'divs', imágenes y links) que son más relevantes en el sitio. Este mapa puede ser usado como un indicador de los principales objetos para compararlos con lo que los diseñadores y administradores quieren como objetos claves.

4. ***Índice de Claridad***

   Este índice muestra la claridad y el orden de una página web. Dicho de otra forma, es la facilidad que tiene el usuario para explorar visualmente un sitio. La literatura aconsejaría tener un índice de claridad alto.

5. ***Mapa de Probabilidad de Click***

   Este mapa muestra qué tan probable es que el usuario haga click en cada una de las opciones que ofrece la página. Esto revela si la página está poniendo énfasis en los caminos que realmente interesan al usuario para seguir navegando en el sitio.

6. ***Mapa de Percepción***

   ¿A dónde se dirige la mirada del usuario en los primeros 3 segundos? Es una pregunta que se responde con este mapa, útil cuando se está teniendo un alto porcentaje de rebote o se quiere entusiasmar rápidamente a futuros usuarios.

1.2. Justificación

El proyecto AKORI considera bastante investigación previa en lo que involucra identificación de objetos web, eye tracking y emocionalidad, sin embargo el proyecto no ha sido capaz de cumplir sus servicios bajo los estándares de precisión deseada, es por esta razón que nace este trabajo de memoria, el trabajo consiste en clasificar páginas web con respecto a su contenido web, es decir, la clasificación considera contenido HTML, diseño, tópico textual y seguridad web, la relación entre la clasificación web y el mejoramiento de AKORI tiene varias aristas.

(a) ***Mejorar desempeño***

   La clasificación se realiza de tal manera que las clases de páginas web son suficientemente distintas entre si y similares dentro de una misma clase, por lo tanto debiera ser más fácil predecir dentro de una misma clase si se conoce que clase es.

(b) ***Listado de objetos web***

   El primer paso para realizar una correcta clasificación web es la parametrización de la página web, esta parametrización rescata todos los objetos web que serán utilizados co-

mo insumo del algoritmo clasificador, lo que además es un importante resultado puesto que si bien este trabajo no dice la posición del objeto web, habla de que objetos son posibles de encontrar dentro de una página web.

(c) ***Segmentación de clientes***

Realizar una segmentación apropiada hace posible diseñar herramientas personalizadas y una vista personalizada (asignar prioridad a distintos indicadores y gráficos) en la página web de resultados.

1.3. Objetivos

Este trabajo de titulo se compone de un objetivo general y cuatro objetivos específicos.

1.3.1. Objetivo General

Diseñar y desarrollar un módulo de clasificación de páginas web en base a las características de su contenido utilizando técnicas de minería de datos.

1.3.2. Objetivos Específicos

1. Realizar un estudio sobre el estado del arte de clasificación en la web y tipos de páginas web.
2. Diseñar, desarrollar e implementar un proceso capaz de parametrizar una página web generando un set de variables cuantitativas.
3. Seleccionar los algoritmos de Data Mining adecuados considerando el juego de datos y variables.
4. Evaluar los resultados de la clasificación web con respecto a trabajos similares.

1.4. Hipótesis de investigación

"Es posible clasificar páginas web en base a las características de su contenido".

1.5. Resultados Esperados

Los resultados esperados tienen directa relación con los objetivos específicos

a) Definir las categorías a considerar de sitios web Chilenos.

b) Definir los parámetros del Vector de Características (Feature Vector).

c) Clasificar páginas web.

d) Validar la hipótesis de investigación.

1.6. Alcances

Se pretende construir un módulo que sea capaz de clasificar páginas web Latinoamericanas, las cuales deben estar indexadas en Google y deben ser accesibles directamente desde el navegador web, es decir, no requiere un sistema de log in.

La finalidad del proyecto es un producto comercial por lo que se le da prioridad a las páginas web con mayor tráfico y si bien se prioriza la correcta clasificación de páginas web en español, este trabajo no se limita a eso puesto que páginas en inglés tienen suficiente relevancia con respecto al tráfico web para ser ignoradas.

El clasificador que se pretende construir tiene la intención de ser utilizado por los servicios de predicción que ofrece AKORI.

1.7. Metodología

A modo de cumplir los objetivos específicos, se propone la siguiente metodología.

La primera parte consiste en un profundo estudio del estado del arte de todos los contenidos relacionados con el proyecto, es decir, categorías de páginas web, clasificación web, web mining, data mining, contenido HTML, web crawling, librerías de programación relacionadas con contenido web, etc.

La segunda parte esta relacionada por una parte con la definición de categorías web, la cual es el producto del estudio realizado en la primera parte y por otro lado el desarrollo de software que permita la creación del vector de características y el juego de datos (etiquetado manual de páginas web), que es la parte crítica de este proyecto, puesto que ese vector de características representa el potencial para que el clasificador tenga posibilidad de tener un buen desempeño.

La tercera parte se procesan los datos resultantes del vector de características junto con el juego de páginas web etiquetadas. Para esto se lleva a cabo el proceso KDD (Knowledge Discovery in

Databases), en el que se aplican técnicas de Data Mining para llevar a cabo la clasificación web y encontrar relaciones no triviales entre las variables.

Finalmente se concluye el trabajo en relación a la hipótesis de investigación y los resultados obtenidos, para luego proponer recomendaciones y trabajo futuro en base al trabajo realizado.

1.8. Estructura del informe

El informe está compuesto por 7 capítulos.

El Capítulo 1 es el capítulo introductorio, en el cual se define el problema, se contextualiza y se plantea el como será resuelto.

El Capítulo 2 corresponde al marco conceptual, que corresponde a todas las definiciones y conceptos necesarios para la correcta lectura de este trabajo de memoria.

El Capítulo 3 explica todo lo relativo a categorización web, lo cual es como se encuentra el estado del arte en cuanto a las clases de páginas web y como se realiza dicha categorización en el presente trabajo.

En el Capítulo 4 se realiza la construcción del vector de características que corresponde a la parametrización de un sitio web, en este capítulo se detalla como se realiza la recolección de todas las variables a utilizar por los algoritmos de minería de datos.

En el Capítulo 5 se realiza todo el proceso de minería de datos para lo cual se sigue la metodología KDD, en este capítulo se detallan todos los algoritmos a utilizar, su calibración y como varia el desempeño con distintos juegos de variables.

El Capítulo 6 muestra en que medida se cumplen los resultados esperados, como se comporta el trabajo en relación al estado del arte y otros resultados importantes que agregan valor al trabajo.

El Capítulo 7 concluye el trabajo, destacando los aprendizajes de este y propuestas de trabajo futuro y recomendaciones.

Capítulo 2

Marco Conceptual

En este capitulo se introducen los conceptos sobre los cuales se enmarca el trabajo de memoria con el objetivo de facilitar la lectura y el entendimiento de esta.

2.1. Internet

Internet remonta sus orígenes a 1969 con ARPANET (Advanced Research Projects Agency Network), una rama militar enfocada en la investigación sin fines comerciales directos, el proyecto consistía en conectar cuatro centros de investigación de tal forma que eran capaces de particionar data en paquetes y enviarlos a través de la red[14].

Actualmente se entiende por internet al sistema interconectado de redes computacionales que intercambian información a través del protocolo TCP/IP, en el año 2016 hay 22.9 billones de dispositivos conectados a internet[15] los cuales acceden a la información a través de la World Wide Web (WWW)[16].

2.1.1. Web

La World Wide Web (WWW), conocida simplemente como "*web*" nace en 1989 como un proyecto de CERN (European Organization for Nuclear Research) y liberada al público en 1993[17].

Se entiende por WWW al espacio en el cual los objetos de interes, conocidos como recursos son identificados por el identificador global llamado URL (Uniform Resource Identifiers), conectado por links de hypertexto y accedido a través de internet[18].

2.1.2. Sitio Web

Un sitio web es una colección de páginas web relacionadas, identificadas por un nombre de dominio común y publicadas en al menos un servidor web. Son accesibles a través de una red IP al referenciar una URL que identifica el sitio web[19].

2.1.3. Página web

Una página web es un documento que puede contener texto, sonido, aplicaciones, programas, vídeo, entre otros. En general se encuentra adaptada a la World Wide Web y navegadores web, puede ser accedida a través de un navegador web a través de su URL sin la necesidad de entrar a la página web de entrada de su sitio web correspondiente[20].

2.1.4. URL

Es el Localizador de Recursos Uniforme (URL debido al acrónimo en inglés), esta formado por una secuencia de caracteres de acuerdo a un formato estándar que designa recursos en la web. El URL se compone de 5 elementos básicos[21].

$$Scheme : [//[user : password@]host[: port]][/]path[?query][\#fragment]$$

$$Ejemplo : http : //www.server.example/xyz/bar/zap.html$$

1. ***Esquema (Scheme)***

   El esquema indica el protocolo de red que se usa para recuperar la información del recurso identificado. Un URL comienza con el nombre de su esquema, seguido por dos puntos, seguido por una parte específica del esquema, los esquemas más comunes son: http, https, ftp, mailto, file, data.

2. ***Autoridad***

   La parte autoridad está compuesta de tres partes.

   (a) Una autenticación opcional que considera usuario y contraseña separados por $''$:$''$ y seguidos por $''$@$''$.

   (b) El host, que consiste en el nombre registrado para un dominio o una dirección IP.

   (c) Opcionalmente se puede ingresar el numero del puerto a utilizar, separado por $''$:$''$ del host.

3. ***Ruta (Path)***

   La ruta contiene data organizada de forma jerárquica, esas separaciones usualmente son una ruta hacia un archivo en el sistema, aunque puede no tener relación alguna.

4. ***Consulta (Query)***

   La consulta es opcional y se encuentra separada de la ruta por el símbolo ″?″ y contiene una consulta de tipo *String* de data no jerárquica cuya estructura es indefinida.

5. ***Fragmento (Fragment)***

   El fragmento se separa de la parte anterior por el símbolo ″#″, este contiene un identificador de fragmento que provee una dirección a un recurso secundario. En general un recurso primario corresponde a un documento HTML y uno secundario a un atributo id de un elemento en específico.

2.2. Web Crawling

Web Crawling es el proceso mediante el cual los motores de búsqueda recolectan paginas desde la web[22], también considera la navegación la web de forma automatizada. Generalmente el web Crawling se realiza con fines de mantener actualizada cierta información web u obtener cierta información. En este trabajo en particular es necesario entender Web Crawling, solo como un proceso anexo a Web Scraping.

2.3. Web Scraping

Web Scraping corresponde al set de técnicas utilizadas para obtener información de un sitio web automáticamente en vez de copiarla manualmente. El objetivo de un Web Scraper es mirar cierto tipo de información, extraer y agregarla en nuevas paginas web. En particular, Web Scrapers se enfocan en transformar data sin estructura y guardarla en bases de datos estructuradas[23].

2.4. Web mining

Se entiende por web mining a la aplicación de técnicas de minería de datos en la web, lo que resulta en la extracción de información valiosa que permite una mayor comprensión sobre el comportamiento de navegación y preferencias de los usuarios de un sitio web.

Dependiendo del tipo de dato a procesar, Web Mining puede ser dividido en tres grandes grupos[24][25]:

1. ***Web Structure Mining (WSM):*** Estudia el análisis de la evolución de la estructura de hipervínculos de un sitio web.

2. ***Web Content Mining (WCM):*** Estudia lo relacionado con el análisis de los contenidos.

3. ***Web Usage Mining (WUM):*** Estudia el comportamiento de los usuarios para conocer un poco más de las navegación de estos en el sitio.

2.5. Vector de Características

Corresponde a un vector n-dimensional de atributos numéricos, en el cual cada atributo representa algún objeto que debería agregar valor al modelamiento posterior, es utilizado comúnmente para reconocimiento de patrones y maquinas de aprendizaje[26].

2.6. Proceso KDD

KDD es el acrónimo de Knowledge Discovery in Databases que hace referencia al proceso no trivial de identificar patrones validos y potencialmente útiles a partir de datos [27].

El término proceso implica que KDD contempla varios pasos, los cuales involucran preparación de data, búsqueda de patrones, evaluación de conocimiento y refinamiento, todos repetidos en iteraciones múltiples. Cuando se habla de que KDD es un proceso no trivial se debe a que siempre se necesita cierto grado de inferencia, los resultados no son directos.

Los resultados producto del proceso KDD (patrones) deben ser válidos en datos nuevos con cierto nivel de confianza, además estos deben al menos tener beneficios potenciales y ser útiles para el usuario final.

Como se menciona anteriormente el proceso KDD se puede entender como una secuencia de pasos a seguir, es correcto también tratar dicho proceso como una metodología, la que se explica a continuación:

1. **Primero:** Desarrollar un entendimiento del dominio de la aplicación, de la relevancia del conocimiento a extraer y identificar el objetivo del proceso KDD desde el punto de vista del cliente.

2. **Segundo:** Es crear un juego de datos objetivo, es decir, seleccionar un juego de datos o en su defecto de un subset de este o simplemente una muestra de datos.

3. **Tercero:** El tercer paso es limpiado y preprocesamiento. Las operaciones básicas de esta etapa comprenden sacar ruido si es apropiado, recolectar información necesaria para modelar, decidir estrategias para tratar valores ausentes y decidir los rangos temporales que se utilizaran.

4. **Cuarto:** Reducción de datos y proyección, se refiere a encontrar características útiles para representar la data dependiendo del objetivo de la tarea, esto considera reducción de dimensionalidad y métodos de transformación (el numero efectivo de variables a considerar puede ser reducido).

5. **Quinto:** Es hacer el cruce entre el objetivo del proceso KDD (paso 1) con un método particular de data mining: clasificación, regresión, clusterización, etc.

6. **Sexto:** Es un análisis exploratorio y selección de modelo e hipótesis, se escogen los algoritmos de data mining y métodos para buscar patrones en los datos, este proceso incluye entender las motivaciones y restricciones del cliente al momento de decidir si es necesario entender las reglas y pesos que atribuye el modelo a las distintas variables y de esta manera tener control sobre el porque el modelo predice de la forma en la que lo hace.

7. **Séptimo:** Es Data mining, es la búsqueda de patrones de interés en una forma de representación particular, esta puede ser: reglas de clasificación, árboles, regresión o *clusters*. El usuario puede ayudar significativamente al método de data mining realizando los pasos anteriores de manera correcta.

8. **Octavo:** es interpretar los patrones rescatados, de no ser experto en el dominio de los datos (ejemplo: salud, negocios, marketing, etc) puede ser recomendable consultar los resultados y conclusiones con un experto o simplemente con el cliente para corroborar si las conclusiones parecieran tener sentido. Posiblemente sea necesario regresar a algún paso entre el primero y el séptimo e iterar. En este paso también se considera la visualización de los patrones rescatados.

9. **Noveno:** Es el uso del conocimiento de forma directa, es la incorporación del conocimiento en otro sistema o acciones futuras o simplemente su correcta documentación, se debe también resolver posibles conflictos con el conocimiento que se creía correcto hasta antes del presente estudio.

   El proceso KDD puede involucrar un numero no menor de iteraciones y en algunos casos generar ciclos infinitos en cualquier par de pasos, el flujo esperado se ilustra en la figura X. Si bien la mayoría de los trabajos se enfoca en el paso siete, los demás pasos son tan importantes como este para realizar un proceso KDD exitoso.

2.6.1. Minería de datos

Existen muchos métodos de minería de datos utilizados para distintos objetivos, la taxonomía se hace cargo de esto y distingue dos grandes tipos de data mining: métodos orientados a la verificación y métodos orientados al descubrimiento[28].

Los métodos orientadas a la verificación tratan la verificación de una hipótesis propuesta por una fuente externa (un experto por ejemplo), estos métodos incluyen algoritmos tradicionales de estadística como fit test, test de hipótesis, ANOVA.

Los métodos orientados al descubrimiento se basan en el aprendizaje inductivo, en el cual

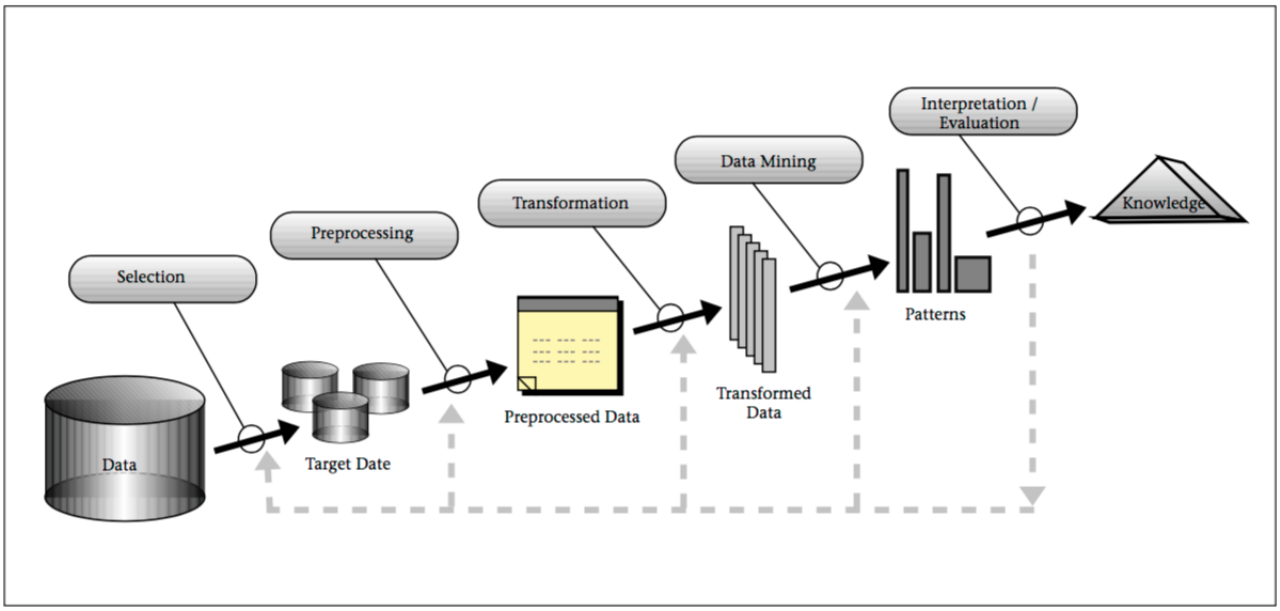


Figura 2.1: Proceso KDD
Fuente:Data Mining to Knowledge Discovery in Databases

un modelo es construido explícitamente o implícitamente de la generalización de un número suficiente de ejemplos de entrenamiento.

Otra distinción importante en la taxonomía de Data mining es dentro de los métodos orientados al descubrimiento, los cuales pueden pertenecer al aprendizaje supervisado (también llamados de predicción), es decir, los que intentan encontrar una relación entre los atributos de entrada (variables independientes) y un atributo objetivo (variable dependiente) y por otro lado se encuentran los pertenecientes al aprendizaje no supervisado (lo cual cubre solo una porción de los métodos de descripción), este tipo de aprendizaje se refiere a técnicas que agrupan observaciones sin una variable dependiente predefinida.

Los métodos supervisados pueden ser implementados en una alta variedad de campos, como marketing, finanzas, manufactura, logística o salud. Es útil realizar una distinción dentro de los modelos que trabajan bajo aprendizaje supervisado, puesto que resuelven problemas distintos, por un lado se encuentran los modelos de regresión, los cuales son capaces de estimar el valor de la variable dependiente en la forma de una variable de *valor real*[1], una de las aplicaciones más comunes de estos modelos son la estimación de demanda, por otro lado existen los modelos de clasificación, los cuales mapean en espacio de los atributos de entrada en clases predefinidas por ejemplo en la clasificación web.

[1] Valor perteneciente al dominio de los números reales

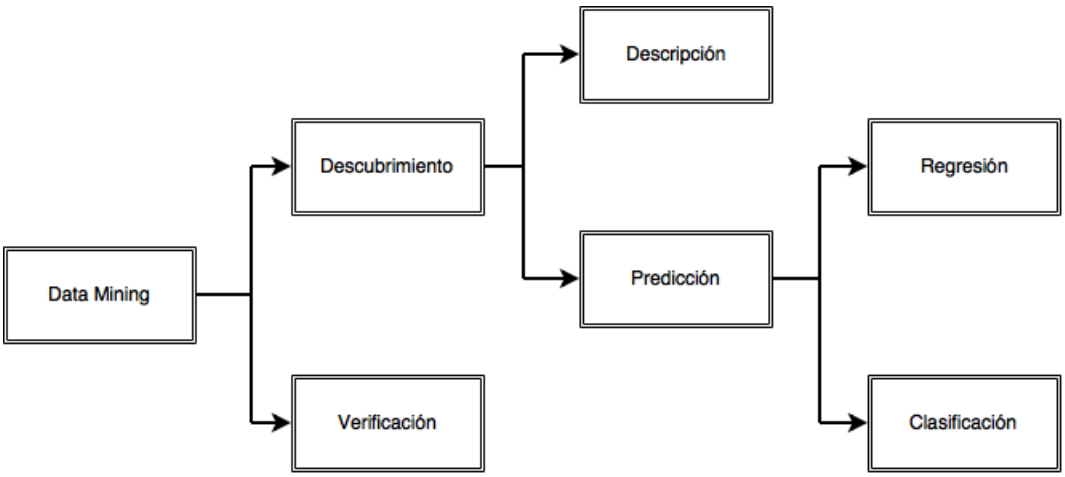


Figura 2.2: Taxonomía Data mining
Fuente: Adaptación de figura de [28]

2.6.1.1. Algoritmos de clasificación

Los problemas clásicos de aprendizaje supervisado requieren mutuamente exclusivas por definición[29], si se considera clasificación en múltiples clases[2] para cada observación pueden existir varias clases candidatas a clasificar dicha instancia pero solo una puede ser la correcta.

Se entregará una breve descripción de los algoritmos de minería de datos más utilizados para la clasificación.

(a) ***Árboles de decisión***

Un árbol de decisión es clasificador expresado como una partición recursiva de una instancia. El árbol de decisión consiste en nodos que forman un árbol arraigado (rooted tree), es decir es un árbol dirigido que posee un nodo padre al cual ningún otro nodo lo precede y se le llama raíz, los nodos que se separan en otros nodos se llaman internos o nodos de prueba, todos los demás nodos son llamados hojas[30].

Cada nodo interno se separa en dos o más instancias dependiendo de la función discreta que define las variables de entrada y cada hoja es asignada a una clase representando el valor objetivo más apropiado, sin embargo una hoja puede contener un vector de probabilidad indicando la probabilidad de pertenecer a una clase.

La probabilidad de pertenecer a una clase se calcula siguiendo las reglas y probabilidades dictadas por todos los nodos de raíz a hoja según los valores de las variables de entrada de la observación como muestra la figura.

Los algoritmos más utilizados para la aplicación de árboles de decisión son los de podado o más conocido como pruning, estos algoritmos se utilizan para hacerse cargo del problema fundamental de los árboles de decisión[31] el cual es el criterio que determi-

[2]Es necesario hacer la siguiente distinción, multiclases: Cada observación puede ser asignada a más de una clase, Clases múltiples: Para cada observación existen más de dos clases candidatos

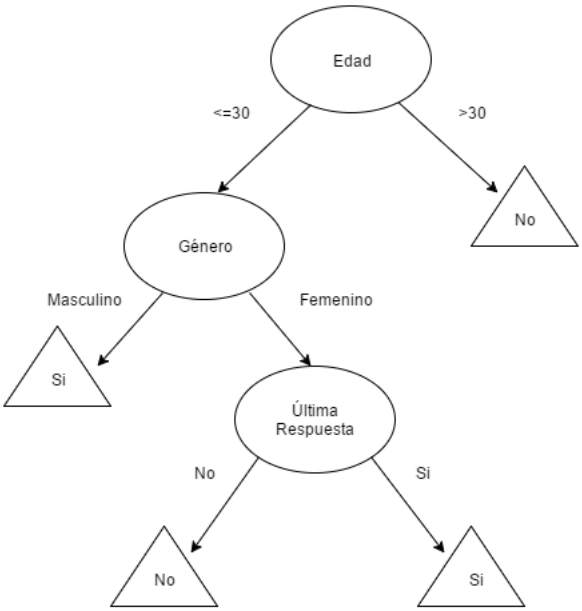


Figura 2.3: Diagrama árbol de decisión
Fuente: Adaptación de figura de [30]

na cuando no deben seguir construyéndose nodos, si el árbol de decisión es muy corto su ajuste sera bajo y tendrá poco poder de predicción y si es demasiado largo estará sobreajustado y tampoco podrá predecir fuera de los datos. El proceso de pruning es un algoritmo recursivo en el cual cada iteración actualiza la función de atributos de entrada con respecto a un criterio en particular (el criterio más común es el de costo) y se repite el proceso hasta que las nuevas particiones no generen una ganancia o se alcance el número dado de iteraciones. El criterio de costo se define de la siguiente forma.

$$\alpha = \frac{\varepsilon(Pruned(T,t),S) - \varepsilon(T,S)}{|Hojas(T)| - |Hojas(Pruned(T,t))|}$$

$T_0, T_1, ..., T_k$ Corresponden a una secuencia de arboles, donde T_0 Es el árbol inicial antes del proceso de pruning.
ε Denota el error del árbol T sobre la muestra S.
$Hojas(T)$ Denota el número de hojas en el árbol T.
$Pruned(T,t)$Denota el árbol obtenido al reemplazar el noto t en T.

(b) ***Bosque Aleatorio***

El Bosque Aleatorio mejor conocido como Random Forest es un algoritmo de clasificación perteneciente a los llamados métodos de aprendizaje por ensamblaje.

En un Random Forest se construyen árboles de decisión utilizando una muestra aleatoria (utilizando la técnica bootstrapping) de los datos, mientras que el árbol de decisión realiza la separación de nodos utilizando la mejor separación posible considerando todas las variables, Random Forest realiza la mejor separación considerando un set de

variables elegidas aleatoriamente, este algoritmo contra intuitivo tiene un desempeño sorprendente en comparación a otros clasificadores[32], ver figura 2.4.

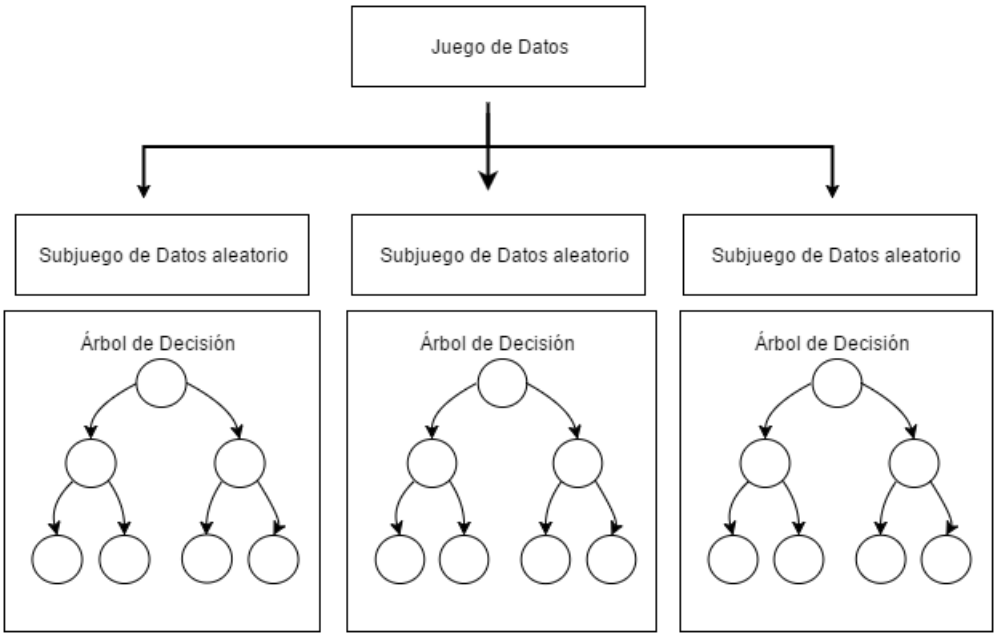


Figura 2.4: Diagrama Random Forest
Fuente: Adaptación de figura de [32]

(c) ***Red Neuronal Artificial***

Las redes neuronales artificiales o ANN (Artificial Neural Network) son modelos computacionales para procesamiento de información y son muy útiles para encontrar para encontrar relaciones entre variables o patrones[33].

Las redes neuronales artificiales heredan su nombre por una simplificación del sistema neuronal biológico, sin embargo mantienen dos de sus características más importantes: procesamiento de información en paralelo y aprendizaje y generalización por experiencia.

Existen tres tipos de redes neuronales comunmente utilziadas: *multilayer feedforward network, Hopfield network, y Kohonen's map*. En este trabajo solo se discutirá la primera debido a que es la más estudiada e implementada y es la que se utilizará para el desarrollo de esta memoria.

Una red neuronal artificial que utiliza *multilayer feedforward network* también conocida como multilayer perceptrons (MLP) consiste en un número de unidades altamente conectadas de computo simple a las cuales se llaman neuronas o nodos. Las neuronas se conectan entre si por arcos, los cuales generan conocimiento y es guardado como peso (weights) segun la robustes de la relación entre las neuronas, si bien cada neurona realiza una tarea simple e imperfecta, como red pueden resolver problemas complejos con resultados notables, en la figura 2.5 se muestra una red neuronal artificial MLP de tres capas.

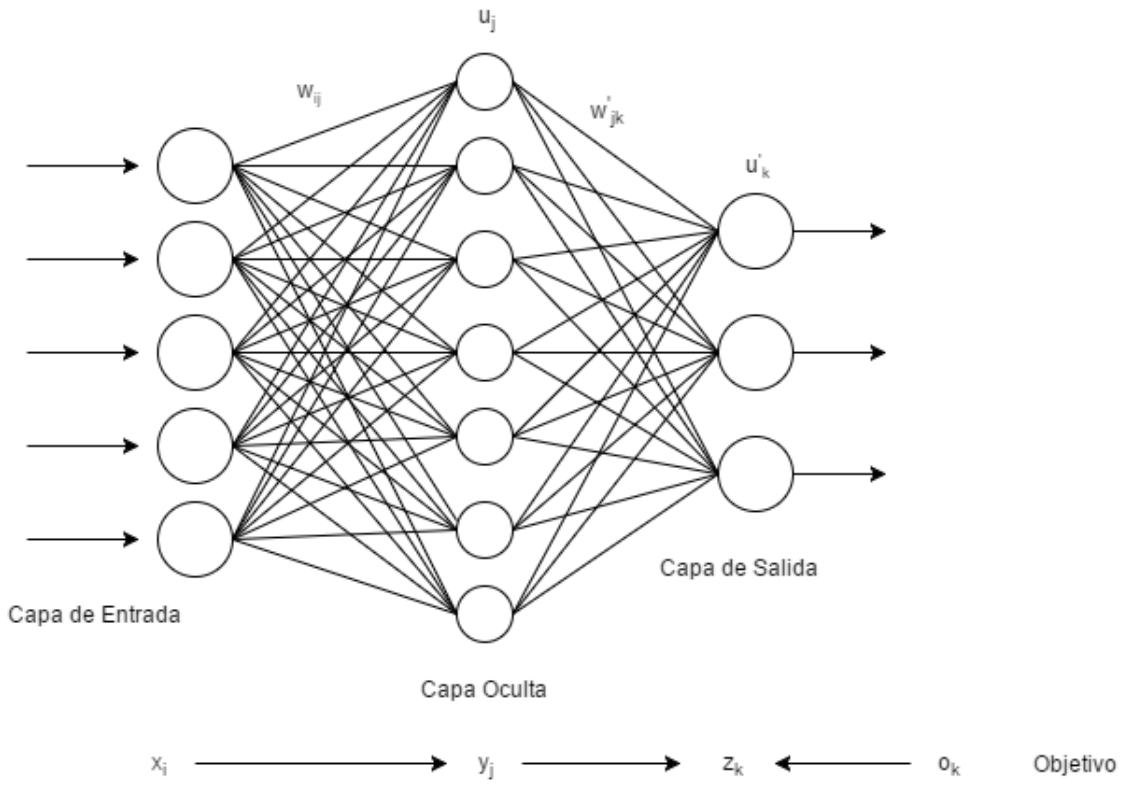


Figura 2.5: Red Neuronal Artificial
Fuente: Adaptación de figura de [33]

Para entender como funciona la matemática de la red neuronal se debe entender que cada neurona procesa información en dos pasos: En el primero los valores de entrada se combinan en conjunto para formar una suma ponderada de valores de entradax_{i} multiplicada por el pesos w_{i} de los arcos que conectan las neuronas y el segundo paso es aplicar una función de transferencia a la suma ponderada antes vista.

$$Resultado_n = f\left(\sum_{\mathrm{i}}^{n} x_{\mathrm{i}} * w_{\mathrm{i}}\right)$$

Comúnmente la función de transferencia puede ser: sigmoide, hiperbólica, seno o identidad. Hasta ahora se ha discutido como se calcula de forma simplificada el resultado de un ciclo en la red neuronal, pero lo que explica los buenos resultados de la red neuronal es el aprendizaje por experiencia lo cual es a su vez generado por ciclos de entrenamiento y actualización de los pesos de los arcos entre las neuronas.

En el caso más simple y utilizado, el error de una red neuronal artificial viene dado por la suma de errores cuadrados y el método de entrenamiento y actualización de pesos más importante y popular es el de *back propagation*, el cual es un método recursivo de gradiente que sigue la siguiente fórmula.

$$w_{\mathrm{i}j}^{nuevo} = w_{\mathrm{i}j}^{viejo} + \Delta w_{\mathrm{i}j}$$

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}$$

Δw_{ij}: Es el gradiente de la función E con respecto al peso w_{ij}.
η: Es la tasa de aprendizaje que controla el tamaño de descenso del gradiente.

Un parámetro adicional que se suele agregar a *back propagation* es *momentum* que es proporcional a la ultima actualización de pesos y de esta forma previene la oscilación entre cambios de pesos anteriores y estancamiento en óptimos locales.

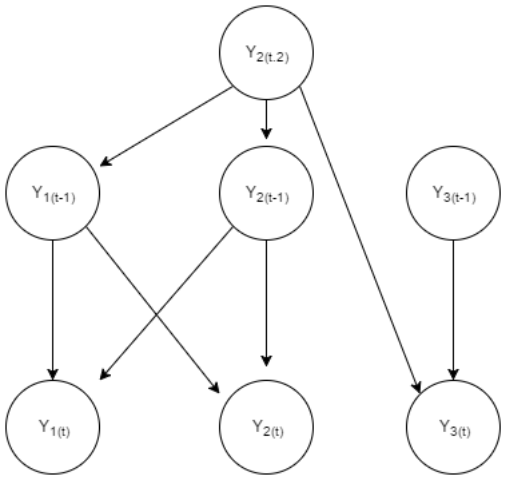
El algoritmo requiere un proceso iterativo que termina cuando no existen ganancia luego de una iteración o se llega al número de iteraciones definido.

(d) ***Redes Bayesianas***

Una red bayesiana es un clasificador que se compone de dos partes: un grafo dirigido acíclico y una distribución de probabilidad. Es posible visualizar una red bayesiana como se muestra en la figura 2.6, el grafo presente sigue las propiedades de una cadena de Markov, en particular que cada nodo es independiente de su nodo no descendiente dado los nodos padres de este, lo que lleva a la directa factorización de la probabilidad conjunta de la red de variables de la siguiente forma[34].

$$p(y_1), ..., y_v = \prod_{i}^{n} p(y_i | pa(y_i))$$

Donde $pa(y_i$ es la variable padre de la variable y_i



		Y3(t)	
Y2(t-2)	Y3(t-1)	-	+
-	-	0.9	0.1
-	+	0.7	0.3
+	-	0.6	0.4
+	+	0.1	0.9

Figura 2.6: Diagrama Red Bayesiana
Fuente: Adaptación de figura de [34]

10. ***Support Vector Machine***

El SVM es un algoritmo de clasificación binaria en su forma estándar y es uno de los más utilizados en maquinas de aprendizaje[35].

Para su implementación se supone un set de entrenamiento $\{(x_1,y_1),(x_2,y_2),...,(x_n,y_n)\} \in \mathbb{R}^N$

en donde x_i es el vector de datos y y_n es el vector de etiquetas de clases que por lo general toma el valor 1 o -1. Support Vector Machine busca la función lineal de la forma de la ecuación.

$$f(x) = \langle w \cdot x \rangle + b$$

de modo que si el vector x_i es de la clase positiva, entonces $f(x_\mathrm{i}) \geq 0$ y para la clase negativa el caso contrario $f(x_\mathrm{i}) \leq 0$, como se muestra a continuación.

$$y_\mathrm{i} = \begin{cases} 1 & si\,\langle w \cdot x + b \geq 0 \rangle \\ -1 & si\,\langle w \cdot x + b \leq 0 \rangle \end{cases}$$

El parámetro w se denomina peso y el parámetro b se denomina sesgo.

El Support Vector Machine busca un hiperplano que sea capaz de separar los datos de entrenamiento entre ambas clases, ese hiperplano es llamo superficie de decisión como se muestra en la figura 2.7.

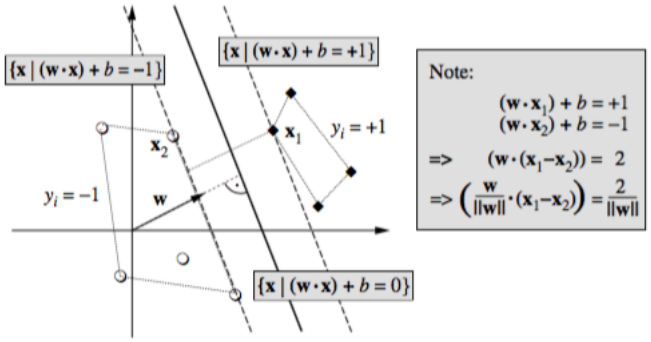


Figura 2.7: Diagrama SVM
Fuente: Adaptación de figura de [35]

El margen se define como la distancia mínima entre un ejemplo de los datos de entrenamiento y la superficie de decisión, luego la función objetivo del algoritmo es maximizar el margen que separa ambas clases, lo que se expresa de la siguiente forma.

$$\text{máx} : \frac{1}{2}\|w\|^2$$
$$s.a. : y_\mathrm{i} \cdot (\langle w \cdot x_\mathrm{i} \rangle + b) \geq 1, \mathrm{i} = 1,...,n$$

La optimización anterior solo se encarga de problemas lineales, para problemas no lineales se utiliza el método de kernel, el cual consiste en hacer calzar el hiperplano de máximo margen en un espacio de características F, siendo F un mapa no linear $\Phi : \mathbb{R}^N \to F$ del espacio de entrada original, generalmente con una dimensionalidad mucho mayor como se muestra en la figura 2.8.

La transformación del Kernel genera el siguiente problema de optimización.

$$\text{máx} : \sum_{\text{i}=1}^{n} \alpha_{\text{i}} - \frac{1}{2} \sum_{\text{i},j=1}^{n} \alpha_{\text{i}} \alpha_j y_{\text{i}} y_j K(x_{\text{i},x_j})$$

$$s.a. : \alpha_{\text{i}} \geq 0, \text{i} = 1, ..., n \sum_{\text{i}=1}^{n} \alpha_{\text{i}} y_{\text{i}} = 0$$

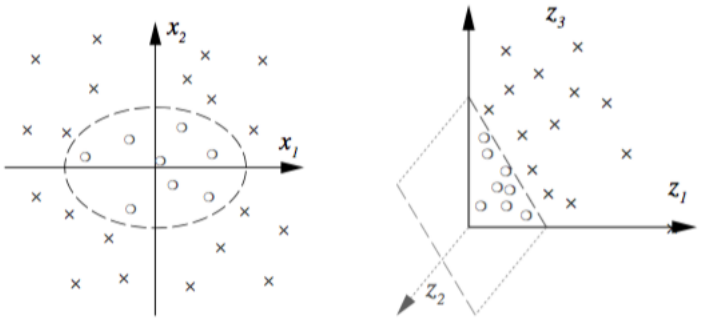


Figura 2.8: Diagrama SVM con Kernel
Fuente: Adaptación de figura de [35]

2.6.2. Minería de textos

La minería de textos se refiere al proceso de obtención de información de alta calidad a partir de textos, esta información puede ser obtenida de diversas fuentes, sin embargo el estado más común de este tipo de data es con falta de estructura[36] que ha sido la razón para el desarrollo de las técnicas y algoritmos que serán discutidos y analizados posteriormente.

En particular para este trabajo se aplicará lo referente a *Information Retrieval* (IR) que tiene relación con el proceso de utilizar ingeniería sobre la literatura con el objetivo de algún tipo de información útil o patrón.

Si bien es difícil clasificar algoritmos de *Information Retrieval*, se puede hacer una diferenciación considerando las aplicaciones que estos tienen[37].

(a) ***Algoritmos de recuperación:*** Es la clase principal de algoritmos de Information Retrieval y se refiere a extraer información de una base de datos de texto

(b) ***Algoritmos de filtro:*** En esta clase de algoritmos el texto plano es el insumo y el resultado es un subconjunto de este que ha sido procesado o corresponde a una versión filtrada de la original.

(c) ***Algoritmos de indexación:*** Corresponden a los algoritmos que tienen por objetivo construir una estructura de datos de tal manera que haga posible una búsqueda rápida en el texto.

Considerando la naturaleza del problema los algoritmos de indexación no generan valor, puesto que la velocidad de la búsqueda de información no tiene relación el objetivo planteado, con respecto a los algoritmos de filtro se utilizan en parte en la etapa de la creación del vector de características pero no se utilizará para la sección de minería de contenido puesto que se intenta rescatar información con respecto a todo el texto visible que se encuentra en un sitio web y con respecto a los algoritmos de recuperación, se relacionan directamente con el problema que se quiere resolver.

Dentro de los algoritmos de recuperación también existen variadas opciones, las cuales a su vez también se pueden separar por clases, a continuación se realizará una pequeña descripción de las clases en consideración:

(a) ***Boolean Retrieval:***

El procedimiento estándar en este tipo de algoritmos es generar consultas de tipo Boolean de modo que los documentos a analizar entreguen resultados de si poseen las características que se consultaron o no. Debido a la construcción del algoritmo requiere mucha habilidad crear consultas que generen un número manejable de resultados producto de la consulta y que a la vez sea interpretable, esto hace necesario además que el desarrollador de este algoritmo no sólo debe ser experto en las necesidades del trabajo si no que también de la temática de los documentos. Es por la razón anterior que se decide no trabajar con algoritmos de tipo Boolean, debido a la variabilidad que existe en la web, es una tarea muy difícil estandarizar una consulta que sea aplicable a todo tipo de páginas web y que además entregue resultados interpretables, del hecho de que un documento coincida o no es demasiado decisivo lo que puede inducir a error.

(b) ***Ranked Retrieval:***

Este algoritmo ordena documentos de acuerdo a que tan relevantes son con respecto a una consulta, se les asigna un puntaje en pares con respecto a (consulta,documento), también de la forma (x,y), esto mide que tan bien describe la consulta al documento, si la consulta es solo un término, el puntaje está dado por la frecuencia del término en el documento. Considerando que asigna un puntaje a cada documento y los documentos son ordenados de más relevante a menos relevante el problema de realizar consultas que entreguen un número manejable de resultados deja de ser un problema, que es la mayor complicación para el otro tipo de algoritmos.

A su vez los algoritmos de recuperación de información bajo el enfoque de Ranked Retrieval pueden ser clasificados con respecto al criterio que se utiliza para elegir el ponderador de sus términos coeficiente de Jaccard o term frecuency.

La gran diferencia es que Jaccard: $Jaccard(A,B) = \frac{A \cap B}{A \cup B}$ solo considera los términos que tienen en común dos documentos con respecto a su extensión, pero no considera la frecuencia de estos, es decir cuantas veces aparecen y por consiguiente no le agrega un valor adicional a términos poco comunes que pueden ser muy importantes para term frecuency, lo cual tiene mucho valor para clasificar páginas web por clases.

Proceso TF-IDF Se entiende por proceso TF-IDF a la metodología de recuperación de información que considera el método de asignación de pesos según TF-IDF. El proceso Term Frecuency - Inverse Document Frecuency (TF-IDF) es uno de los algoritmos para análisis de textos más comunes[38] el cual cuenta con 7 etapas, las que se detallan a continuación adaptados a la web.

i. ***Parsing***

Es el proceso de extracción de todo el texto visible que contiene una página web, se eliminan todos los Tags HTML, se transforman las mayúsculas a minúsculas, se eliminan tildes, puntuación, comillas y todo signo lingüístico que no sean letras.

ii. ***Tokenización***

Dentro de este proceso se transforma el texto en palabras, es decir, el texto es transformado en un vector en el cual cada parámetro es una palabra, el criterio para la división de palabras es el espacio " ".

iii. ***Stopwords***

En este proceso de eliminan las palabras que no contribuyen a la semántica[3] del texto, generalmente son las palabras comunes del idioma.

iv. ***Stemming***

Se refiere al proceso de identificación de semántica, trata de agrupar palabras que poseen el mismo significado en una sola, por ejemplo: Conectar, Conectado y Conectando serian reemplazas por la misma palabra: "Conectar".

v. ***Generación de n-gramas***

Considerando que se tiene una lista limpia de palabras que pasaron por el proceso de *stemming*, se procede a someterlas a una ultima transformación, la creación de bi-gramas, puesto que des-ambigua palabras que no fue posible con palabras singulares y los tri-gramas no agregaban valor por sobre los bi-gramas.

vi. ***Word Page Vector***

Se calcula la frecuencia de cada palabra (o bi-grama) y luego el Inverse Document Frecuency:

$$TF - IDF = TermFrecuency_{ij} * \log(Q/n_i)$$

i: Identificador de Término lingüístico.
j: Identificador de página web.
Q: Número de páginas.

[3]Se entiende por semántica al estudio del significado en un proceso de comunicación

vii. ***Data mining***

Para finalizar este proceso se implementan algoritmos de Data mining similares a los utilizados en aprendizaje supervisado y clasificación.

En la figura 2.9 se muestra el resumen del proceso de obtención del vector de características relativo a Information Retrieval.

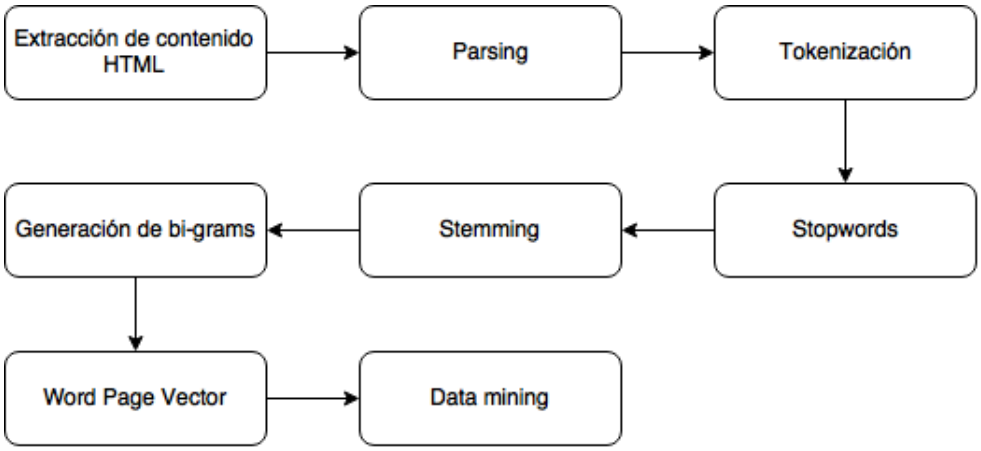


Figura 2.9: Proceso TF-IDF
Fuente: Elaboración propia

2.6.3. Evaluación de clasificadores

Existen variados evaluadores de clasificadores al momento de evaluar un modelo, a continuación se realizará una descripción de los más importantes.

Con el objetivo de facilitar el entendimiento, se utilizará una matriz de confusión normalizara para ejemplificar las métricas, figura 2.10.

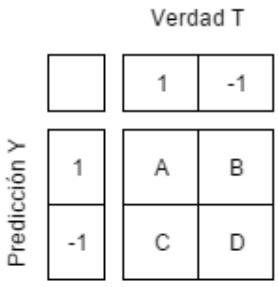


Figura 2.10: Matriz de confusión
Fuente: Elaboración propia

Accuracy: Fracción total de instancias correctamente clasificadas.

$$Accuracy = A + D$$
$$Accuracy = P(Y = 1, T = 1) + P(Y = -1, T = -1)$$

Precision:Fracción de instancias clasificadas como positivas que la realidad son positivas.

$$Precision = \frac{A}{A + B}$$
$$Precision = P(T = 1|Y = 1)$$

Recall: También conocida como sensibilidad, corresponde a la fracción de las instancias positivas que fueron predichas positivas.

$$Recall = \frac{A}{A + C}$$
$$Recall = P(Y = 1|T = 1)$$

F-Score: Corresponde a la media armónica entre *precision* y *recall*.

$$F - Score = 2 \cdot \frac{Recall \cdot Precisin}{Recall + Precisin}$$

2.6.4. Métodos de validación

Validación por división

La validación por división o Split validation es una técnica de evaluación de modelos de clasificación que consiste en particionar aleatoriamente la data según una proporción definida, convencionalmente se realiza una partición de 80 % de la data para entrenar el modelo y 20 % de la data para testear. Este tipo de validación es cada vez menos común por su dependencia de los resultados con respecto a la partición[39].

Validación cruzada

La validación cruzada, también conocida como validación X, es una técnica utilizada para evaluar resultados estadísticos, esta técnica tiene la ventaja sobre otros evaluadores en que garantiza que los resultados son independientes de la partición que se utiliza entre los datos de entrenamiento y de prueba. Consiste en repetir y calcular la media aritmética obtenida de las medidas de evaluación sobre diferentes particiones. Se utiliza en entornos donde el objetivo principal es la predicción

y se teme sobreajustar el modelo con los datos[40].

2.7. Arquitectura de seguridad

Se entiende por arquitectura de seguridad según IETF Internet Security Glossary publicada en RFC 2828[41] por los servicios de seguridad que requiere un sistema para cumplir las necesidades de sus usuarios y poseer los niveles de desempeño necesarios en los elementos para tratar con las amenazas del medio.

En el mundo digital se debe tener un conocimiento claro de que es lo que se intenta diseñar y cuales son los posibles adversarios se deben tener en cuenta, con que recursos cuentan (en términos de poder computacional y tiempo), cuales son sus posibles estrategias y cuales son los costos de su éxito. Todo lo anterior debe estar reflejado en un análisis de amenaza y riesgo en el momento de la creación de la arquitectura.

Una arquitectura de seguridad distingue entre cinco clases de servicios de seguridad: Autenticación, control de acceso, confidencialidad de data, integridad de data y "non-repudiation"(sin repudio)[42].

1. ***Autenticación***

   Como su nombre lo sugiere corresponde al servicio que provee la autenticación de una entidad o origen de datos.
   La autenticación de una entidad corresponde a verificar que esta entidad es realmente quien dice ser, la autenticación de origen de datos es la verificación de la fuente en donde se generan dichos datos es efectivamente la que se expone, sin embargo este servicio no provee protección contra duplicación o modificación de esta en el proceso de transferencia.
   Este proceso es prerrequisito de los servicios de provisión de autorización y control de acceso.

2. ***Control de acceso***

   Este servicio existe para proteger los recursos de sistema contra uso sin autorización.
   El uso de recursos de sistema se define sin autorización cuando la entidad que desea hacer uso de estos no tiene los permisos o privilegios necesarios para hacerlo, es por esto que los servicios de control de acceso y autenticación están muy unidos y bajo ciertos conceptos pueden ser referidos como un servicio que los involucra a ambos.

3. ***Confidencialidad de datos***

   La confidencialidad de los datos se refiere a denegar acceso a entidades sin autorización. Existen muchas formas de entregar este servicio, a continuación se nombrarán las mas comunes:

   (a) Confidencialidad para todos los datos transferidos a través de una conexión.

(b) Confidencialidad para unidades de datos individuales.

(c) Confidencialidad para ciertos campos dentro de un juego de datos para unidades de datos individuales o toda la data en una conexión.

Estos servicios de confidencialidad de datos se pueden implementar de manera relativamente simple utilizando técnicas criptográficas estándares.

4. ***Servicios de integridad de datos***

   La integridad de los datos se refiere a la propiedad de no permite que la data sea alterada o destruida de manera no autorizada. Existe más de una forma de lograr este servicio:

   (a) Integridad de datos con recuperación: Integridad para todos los datos transferidos en una conexión y si es posible, la perdida de integridad es recuperada.

   (b) Integridad de datos sin recuperación: Similar al servicio anterior, pero la perdida de integridad no es recuperable.

   (c) Integridad para ciertos campos de un juego de datos dentro de una misma conexión.

   (d) Integridad para datos para unidades individuales de datos.

   (e) Integridad para ciertos campos de un juego de datos para unidades individuales de datos.

   El uso de autentificaron al inicio de una conexión y el servicio de integridad de una conexión proveen en conjunto corroboración de que la fuente de todas las unidades de datos transferidos, la integridad de esas unidades de datos y que no existen duplicados.

5. ***Servicio de sin repudio***

   Este servicio es implementado para prevenir que entidades involucradas en una comunicación puedan negar haber participado en dicha comunicación. Este servicio puede ser implementado tanto en la parte del emisor, receptor o ambos, enviando pruebas de envió y recepción exitosa.

   Este servicio adquiere real importancia en negocios basados en internet, los conocidos E-Commerce y de esta manera no se puede denegar un pago luego de haber comprado un articulo en algún sitio web.

El término SSL (Secure Sockets Layer) o Capa de Conexiones Seguras, es un término propuesto por Netscape comunications en 1990, el cual evolucionó a SSL 2.0, SSL 3.0 y finalmente a TSL (Transport Layer Security) que es el término que se usa en estos días. El protocolo SSL, es un protocolo Cliente/Servidor que provee en su version básica tres servicios fundamentales:

1. Autenticación

2. Conexion Confidencial

3. Conexion de integridad de Servicios (sin recuperación).

Hay que considerar que a pesar de que el protocolo SSL utiliza llaves públicas criptográficas no ofrece ningún tipo de servicio de sin repudio, lo cual es un gran contraste con respecto a las firmas HTTPS o XML que fueron específicamente diseñadas para proveer dicho servicio.

Considerando que SSL es un protocolo orientado a conexiones, las firmas digitales son otorgadas a toda o nada de la data, es decir, no se puede segmentar. El término SSL puede ser visto como una capa intermedia entre la capa de transporte y aplicación para establecer comunicaciones seguras entre entidades.

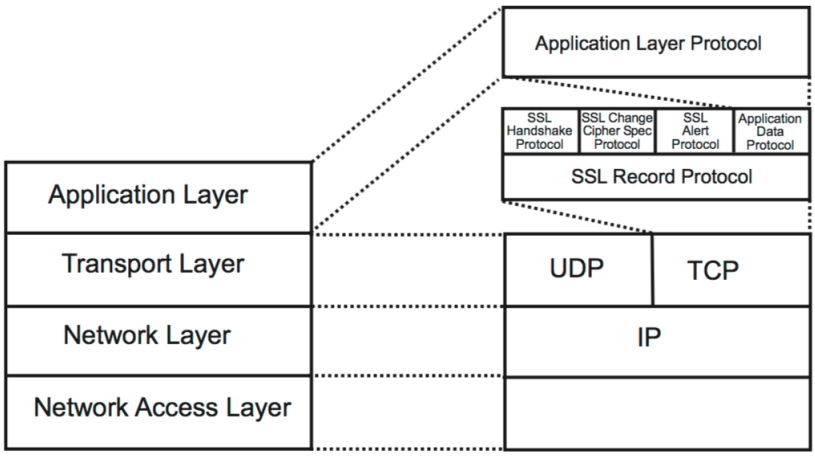


Figura 2.11: Capas y protocolos SSL
Fuente: SSL and TSL theory and practice

2.7.1. Certificado de seguridad X.509

Un certificado X.509 es parte clave del protocolo TSL[43], este certificado especifica el formato de las llaves públicas, vigencia, atributos y ruta de validación. Este certificado también provee información importante tanto de la organización que adquiere el certificado como de la que lo emite.

1. ***CN: Common Name***

   Corresponde al nombre que se le asigna tanto al sitio web como al emisor de certificados. En el caso del sitio web generalmente corresponde a la URL.

2. ***O: Organization***

   Corresponde al nombre de la organización en cuestión, aplica para el sitio web y para el emisor de certificados.

3. ***OU: Organization Unit***

   Corresponde a la unidad dentro de la organización que es responsable de la gestión y obtención de certificados, generalmente tiene relación con el área de tecnologías de información o

business inteligence.

4. ***L: Locality***

   Locality o localidad se relaciona con un área geográfica en donde se encuentra la organización, en el caso particular de Chile la localidad se entiende como la comuna en la que se encuentra.

5. ***ST: State Name***

   Corresponde al estado en donde se encuentra la organización, en el caso particular de Chile, State Name se relaciona con la región en donde se desempeña.

6. ***C: Country***

   Corresponde al país en donde se encuentra físicamente la organización.

   Una parte aun más importante del certificado X.509 son sus extensiones, que es la cual como se menciona anteriormente provee información crítica sobre el protocolo TSL.

2.7.1.1. Extensiones de un certificado de seguridad

Las extensiones del certificado son mecanismos de apoyo que entregan especificaciones TSL, estas especificaciones tienen una forma genérica provista por la RFC 4366[4], la cual viene acompañada del CLIENTHELLO message por defecto[44]. De ser necesario se puede generar una versión extendida mediante la creación de un EXTENDED CLIENTHELLO message, lo cual es un CLIENTHELLO message con un bloque adicional de data, cuya única restricción es no romper los servidores TSL existentes, lo cual sucede cuando el servidor TSL no es capaz de entender las extensiones correctamente.

Un punto importante a considerar es que las extensiones entregan una métrica de criticidad, la cual es una medida de vulnerabilidad, una extensión con criticidad positiva significa que dicha especificación puede ser objetivo de una amenaza.
A continuación se detallaran las extensiones más comunes según el certificado X.509.

1. ***key Size***

   Corresponde al tamaño de la llave pública, el tamaño estándar de una llave de seguridad es de 2048 bits.

2. ***Expiracy***

   Es una variable binaria que representa si la página web alguna vez tuvo certificado de tipo SSL, se considera que existe una diferencia entre páginas web que nunca han tenido seguridad SSL y otras que han optado por no renovarla.

[4]RFC (Request for commons: Es un tipo de publicación de Internet Engineering Task Force (IETF) y the Internet Society (ISOC) las cuales son los trabajos de desarrollo técnico más importantes en el cuerpo de internet.

3. ***AuthorityInfoAccess***

   La extensión authority information access indica como acceder a la información y servicios por entidades empoderadas para hacerlo (derechos). Estos servicios incluyen validación de servicios y políticas con respecto a CA [5]

4. ***AuthorityKeyIdentifier***

   Esta extención es responsable de identificar al Certificate Authority que firmó el certificado, esto se realiza con el objetivo de agilizar el proceso de validación y evitar corroboraciones múltiples.

5. ***BasicConstraints***

   Esta extensión se utiliza para validar que un certificado es un certificado validado por CA. Notar que existen entidades que emiten sus propios certificados (CA privados) y si bien son válidos y legítimos, se considera una vulnerabilidad por defecto.

6. ***CertificatePolicies***

   Los valores y propiedades de esta extensión son muy amplios, todos los campos de esta extensión pueden ser utilizados si su sintaxis es correcta, para lograr el objetivo anterior las recomendaciones PKIX son suficientes. Ejemplo de lo que se puede lograr utilizando la extensión "CertificatePolicies" es el parámetro userNotice que a su vez tiene los parámetros explicitText, organization y noticeNumbers. explicitText y organization son text strings, noticeNumbers es una lista separada por comas (CSV), esto además ejemplifica la variabilidad que existe en esta extensión.

7. ***KeyUsage***

   Esta extención es una variable multivalor, se pueden usar muchos tipos de llaves. En este momento existen los siguientes tipos de KeyUsage: digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment, keyAgreement, keyCertSign, cRLSign, encipherOnly and decipherOnly.

   También existe una extención llamada ExtendedKeyUsage que permite el uso de tipos de llaves adicionales, sin embargo los descritos anteriormente son los por defecto y por consiguiente los más usados.

8. ***SubjectAlternativeName***

   La extension SubjectAlternativeName permite incluir cierta información al archivo de configuración, esto incluye email, URI (uniform resource indicator), DNS (a DNS domain name), RID (Registered ID), IP (IP address), dirName (Distinguished name) y otherName.

[5] CA: Autoridad de certificados, es una entidad de confianza responsable de emitir y revocar certificados de seguridad de firma electrónica.

9. ***SubjectKeyIdentifier***

   Esta extensión es del tipo String, la cual puede adoptar solo dos valores, uno es la palabra ”hash”, la cual indica que se seguir automáticamente el protocolo descrito en la RFC3280 o el segundo valor posible es un String hexadecimal. Se recomienda fuertemente utilizar el valor hash.

2.7.2. Metodología de Rating según seguridad web por SSL LABS

La seguridad web de un sitio web se da a conocer a través nota que es simplemente una transformación de un puntaje hacia una letra, esto corresponde a una asignación lineal. La nota es la suma ponderada de tres categorías:

1. ***Compatibilidad de Protocolo (30%)***

2. ***Intercambio de llaves (30%)***

3. ***Intensidad de cifrado (40%)***

Antes de abordar las categorías descritas anteriormente, es necesario hablar de un tema muy particular: Inspección de certificados, los certificados de un servidor son por lo general el punto más debil dentro de una configuración SSL, un certificado poco confiable (no firmado por un CA confiable), no garantiza que no exista la posibilidad de MITM (Man in the middle), que significa que aunque la configuración SSL sea óptima no garantice que la comunicación no sea exclusiva entre el servidor y el usuario. La seguridad SSL tampoco puede ser garantizada para sitios web que posean certificados expirados.

A continuación se listan problemas de certificados suficientemente graves que resultan automáticamente en un punta je de cero y por consiguiente la asignación de la letra F en lo que respecta a seguridad SSL.

1. Dominio incorrecto.

2. Certificado no válido.

3. Certificado auto asignado.

4. Certificado no confiable (CA desconocido).

5. Certificado revocado.

6. Firma de certificado poco segura (MD2 o MD5).

7. LLave insegura.

2.7.2.1. Protocolo compatibilidad

Un servidor puede tener compatibilidad más de un protocolo, varios con vulnerabilidades conocidas, sin embargo algunas más importantes que otras, de esta manera el puntaje de un servidor se calcula como el puntaje que obtiene el mejor protocolo compatible sumado al puntaje que obtiene el peor y luego esa suma se divide en dos.
A modo de ejemplo:

Protocol	Score
SSL 2.0	0%
SSL 3.0	80%
TLS 1.0	90%
TLS 1.1	95%
TLS 1.2	100%

Figura 2.12: Protocol Support Rating Guide
Fuente: SSL LABS

2.7.2.2. Intercambio de llaves

El intercambio de llaves se realiza en dos etapas, la primera es el proceso de autenticación, de manera que sea posible la correcta asignación de permisos, este intercambio de llaves sucede específicamente al momento de verificar que una entidad sea quien dice ser. La segunda etapa es en el intercambio de llaves secretas que se utiliza en el inicio de sesión y llenado de datos personales.

Si se ignora el proceso de autenticación se es completamente vulnerable a ataques de MIYM, lo que resulta en un puntaje cero en Intercambio de llaves.

La mayoría de los servidores utilizan criptografía pública para el intercambio de llaves bajo los algoritmos ephemeral Diffie-Hellman key exchange (DHE) y Elliptic Crypto variation ECDHE. Esto se traduce en que si la llave privada del servidor es más fuerte, será más difícil de romper el intercambio de llaves. A continuación se detalla la asignación de puntaje:

2.7.2.3. Intensidad cifrado

Para interceptar una comunicación se puede intentar romper el cifrado simétrico que se utiliza para realizar dicha comunicación, un cifrado más robusto requiere un mayor esfuerzo para vulnerar. Debido a que un mismo servidor puede utilizar cifrados de muchas intensidades para calcular el puntaje del cifrado se utiliza un algoritmo similar al utilizado en compatibilidad de protocolos, se suma el puntaje del cifrado de mayor intensidad con el de menor intensidad y se divide en dos.
A continuación se detalla el puntaje asignado a las distintas intensidades de un cifrado.

Key exchange aspect	Score
Weak key (Debian OpenSSL flaw)	0%
Anonymous key exchange (no authentication)	0%
Key or DH parameter strength < 512 bits	20%
Exportable key exchange (limited to 512 bits)	40%
Key or DH parameter strength < 1024 bits (e.g., 512)	40%
Key or DH parameter strength < 2048 bits (e.g., 1024)	80%
Key or DH parameter strength < 4096 bits (e.g., 2048)	90%
Key or DH parameter strength >= 4096 bits (e.g., 4096)	100%

Figura 2.13: Intercambio de llaves
Fuente: SSL LABS

Cipher strength	Score
0 bits (no encryption)	0%
< 128 bits (e.g., 40, 56)	20%
< 256 bits (e.g., 128, 168)	80%
>= 256 bits (e.g., 256)	100%

Figura 2.14: Intensidad de cifrado
Fuente: SSL LABS

Finalmente se realiza una suma ponderada como se describe al inicio de esta sección, es decir Compatibilidad de Protocolo (30 %) + Intercambio de llaves (30 %) + Intensidad de cifrado (40 %) y luego dependiendo del puntaje se le asigna una letra al servidor en cuestión.
A continuación se presenta el equivalente entre puntaje y letra.

Numerical Score	Grade
score >= 80	A
score >= 65	B
score >= 50	C
score >= 35	D
score >= 20	E
score < 20	F

Figura 2.15: Transformación entre puntaje y letras
Fuente: SSL LABS

Tanto la metodología como la asignación de puntaje se encuentra actualizados a 14 de Octubre de 2015 y se obtienen de *SSL Server Rating Guide*[45].

Capítulo 3

Categorización Web

La clasificación web, también conocida como categorización web es el proceso de asignar a una página web una etiqueta predefinida[46]. Este proceso es fundamental para este trabajo de título, su correcta elección permite que las páginas web dentro de una misma categoría compartan la mayor cantidad de características posibles, lo que ayudaría de forma considerable al proyecto final AKORI, es necesario considerar que a medida que aumenta el número de clases mayor será la similitud entre las páginas de una misma clase, sin embargo dificulta la clasificación y cada clase entrega menos valor al trabajo, puesto que será muy similar a otras y posiblemente no tendrá características excluyentes.

El proceso de asignación de etiquetas corresponde a un problema de minería de datos, el cual será abordado en el capítulo 5.

El problema de clasificación web puede ser abordado por varios enfoques, a continuación se describen los más relevantes.

1. ***Clasificación por Tópico:*** Las páginas web se clasifican con respecto a la temática, a grandes rasgos se clasifica una página web dependiendo a que tema se asemeja. Por ejemplo existen los tópicos: “Arte”, “Deporte”, “Negocios”, entre otros.

2. ***Clasificación por Función:*** Las páginas web se clasifican con respecto al rol que juega la pagina web, con respecto a la función que cumple y qué desea satisfacer[47]. Por ejemplo “Pagina de Admisión”, “Pagina Personal”, “Pagina para buscar empleos”, entre otras.

3. ***Clasificación por sentimientos:*** Las páginas web se clasifican con respecto a la opinión del autor sobre el tema especifico a tratar, es una clasificación basada en la opinión. Debido a que es interpretativa (acerca de lo que se infiere que opina el autor), es considerada una clasificación menos robusta[48].

4. ***Clasificación por genero:*** Las páginas se clasifican con respecto a genero de forma más compleja, esta considera además del contenido, la forma, estilo y audiencia[49]. Ejemplos de etiquetas de clasificación por genero son: “Educacional”, “Geek”, “Privado”.

El criterio de clasificación que se debe escoger debe ir ligado a los objetivos que se persiguen

en la linea de investigación, el cual es mejorar el desempeño de un sistema que predice el comportamiento ocular, y considerando la hipótesis validada por Bing Pan[50]: “El comportamiento ocular varía con respecto a la tarea a realizar en su sitio web”, lo que puede extrapolarse a que el comportamiento ocular varia con respecto al servicio que ofrece una página web si se asume que los usuario utilizan las páginas web para satisfacer el servicio que estas ofrecen.

Dado que comportamiento ocular de un individuo no es independiente de su objetivo en la página web, no solo se deben conocer los objetos presente en esta y su contenido web, si no que se debe conocer el porque el usuario se encuentra navegando. Considerando lo anterior se desea realizar una clasificación que permita clasificar los servicios que ofrecen las distintas páginas web, a esta clasificación la llamaremos categorización por servicio.

3.1. Categorización por Servicio

El objetivo de esta categorización es ser capaz de clasificar todo tipo de pagina web Chile en una etiqueta única y excluyente. Se utiliza como base la categorización y definiciones propuestas por Monideepa Tarafdar & Jie Zhang[51], a la cual se le agregan las categorías de Redes sociales y se separan las categorías de noticias e información.

1. ***Motores de búsqueda***

   Los motores de búsqueda, también conocido como "buscadores", son sistemas informáticos que producto de una consulta generada por el usuario, buscan y dan acceso a archivos almacenados en servidores web[52]. El objetivo de un sitio de motor de búsqueda es ayudar al usuario a llegar al sitio web deseado mediante palabras clave.
   Ejemplo `www.google.cl`

2. ***E-Commerce***

   Los sitios web categorizados en esta categoría tienen por objetivo la comercialización de bienes y servicios. Notar que si bien hay una diferencia significativa en el diseño en los distintos tipos de E-Shopping:

   (a) Listing:

   Los objetos se presentan a modo de lista, en la cual cada objeto presenta algún tipo de descripción general, precio y un link a una descripción al detalle.
   Ejemplo `www.mercadolibre.cl`

   (b) Retail

   Los sitios web de retail presentan sus productos mediante mosaicos, con una pequeña descripción, precio, y un link a la descripción al detalle, los objetos son agrupados por categorías y poseen carros de compra.
   Ejemplo: `www.ripley.cl`

(c) cupón

Los sitios web de cupones tienen estructura similar al retail, sin embargo es necesario diferenciarlo debido a que el comportamiento del usuario difiere al del retail lo suficiente como para ser una sub-categoría distinta, la oferta apela al consumismo liderado por las rebajas, a diferencia de por encontrar el objeto que se busca, que es lo que sucede en el retail.

Diferencias sutiles pueden ser encontradas en la estructura:

i. Ausencia de carro de compra.

ii. Menor cantidad de productos.

iii. Menor cantidad de servicios.

iv. Menor cantidad de métodos de pago.

Ejemplo: `www.groupon.cl`

(d) Empresas B2B Los sitios web B2B (Business to Business), son sitios web dedicados a ofrecer productos y servicios a empresas, por lo que generalmente no ofrecen una propuesta pública, esto se debe a que son soluciones a medida y el valor de esta depende del cliente. Una página web de este tipo se caracterizan por realizar una descripción detallada del servicio que realizan, mostrar sus clientes más notorios, generalmente no se muestran precios (pueden estar sujetos a características propias del cliente) y la información de contacto suele ser de fácil acceso.
Ejemplo: `www.ibm.cl`

3. ***Entretenimiento***

El objetivo primario de estos sitios web es mantener la atención e interés de los usuarios o simplemente brindar placer, generalmente esto se logra a través de contenido relacionado a películas, música y juegos.
Ejemplo: `www.lapatilla.com`

4. ***Noticias***

El objetivo primordial de estos sitios web es proveer información de eventos actuales y tópicos específicos como clima, trabajos, bienes raíces.
Ejemplo: `www.emol.com`

5. ***Informativas***

El objetivo fundamental de estos sitios web es enseñar e informar sobre un hecho o actividad en particular, en esta categoría se clasifican sitios que pretender ser tutoriales, foros, descriptivos y información pura.
Ejemplo: `www.stackoverflow.com`

6. ***Servicios financieros***

   Estos sitios web tienen relación con bancos y entidades financieras, se destacan por ofrecer facilidades al momento de realizar trading financiero prestamos de dinero, compra/venta de acciones y flujo de capital en general.
   Ejemplo: `www.Santander.cl`

7. ***Red Social***

   La introducción de la Web 2.0 ha potenciado en gran magnitud la relevancia de estos sitios, llegando a ser uno de los pilares fundamentales en la comunicación web. Las redes sociales son una estructura social web compuesta por actores (usuarios) que se relacionan a través de un criterio y son capaces de comunicarse y compartir contenido.
   Ejemplo: `www.Twitter.com`

3.2. Enfoques para Clasificar Páginas Web

Clasificar un sitio web no es tarea fácil, se pueden utilizar tantas variables como estime el analista, es por esto que existen enfoques al momento de realizar esta labor y así realizar un trabajo consistente.

3.2.1. Clasificación por Texto URL

Este enfoque de clasificación utiliza como insumo solo la dirección web, generalmente se realiza un parseo de la dirección y mediante técnicas de text mining, se es capaz de clasificar cada parte del nombre de la dirección de la pagina web en alguna categoría[53], luego se le asigna un peso a cada una de las partes descritas y se etiqueta con cierta probabilidad.

Este enfoque de clasificación es el que consume menos recursos, lo que lo hace una buena herramienta al momento de enfrentar la web, puesto que la cantidad de paginas web es muy voluminosa, sin embargo debido a que solo considera URL y nada de contenido, no obtiene los mejores resultados si la dirección web no es un nombre sugerente, como por ejemplo una marca (`www.Ripley.cl`).

3.2.2. Clasificación por Contexto

Este tipo de clasificación utiliza como herramienta como se apuntan las páginas entre si a través de hipervínculos, utilizan lógica de algoritmos similares a los que usados por los motores de búsqueda [7] (Ej: PageRank), la premisa es que sitios web generalmente apuntan con mayor frecuencia a sitios web similares.

3.2.3. Clasificación por Contenido

Este tipo de clasificación web solo se centra en el código fuente de una pagina web, desde el cual puede rescatar información con respecto a la estructura del sitio[54], la temática de sus palabras, la seguridad que este posee, analizar las imágenes, colores, etc.

La lógica de este tipo de clasificación es entender como el usuario es capaz de reconocer un sitios web al enfrentarse a el y replicar esa lógica automatizándola.

3.3. Clasificación propuesta

Considerando las tres alternativas propuestas anteriormente se realizará una clasificación por contenido, por las siguientes razones.

1. Considera los objetos web dentro de una página web lo que entrega un listado de dichos objetos que pueden ser utilizados por el proyecto AKORI.

2. Considera el tópico de una página web, lo que debiese obtener mejores resultados si se considera todo tipos de páginas web.

3. El diseño de una página web es lo que debiese tener el mayor impacto al momento de predecir el comportamiento ocular de un usuario, por lo que debiese ser un excelente punto de partida si se quiere emular el çomo"los usuarios son capaces de diferenciar clases de páginas web.

3.4. Vector de Características

El vector de características se crea a partir de las variables propias de una página web que nos permiten a nosotros como usuarios reconocer a que tipo de página web nos enfrentamos y de esta manera ser capaz de identificar variables excluyentes y de esta forma reconocer los objetos clave en dicha página web. El vector de características se crea bajo un enfoque de clasificación por contenido. El vector cuenta de dos partes fundamentales: Contenido Web y tipo de Seguridad

3.4.1. Contenido Web

Esta es la parte más importante del vector de características, este proceso se realizar a través de un Web Scrapper, el cual consiste en un Módulo desarrollado en Java que se conecta a una URL, que apunta hacia una o varias páginas de un sitio web y extrae contenido clave (definido por el desarrollador), se rescata tanto contenido visual para el usuario como contenido oculto, como por ejemplo, contenido relacionado a la descripción del sitio web <meta name = *“description”* content = *“Crea una cuenta o inicia sesión en Facebook. Conecta con amigos, familiares y otras personas*

que conozcas. Comparte fotos y vídeos, envía mensajes y...”>. Cada consulta realizada es guardada como una variable en el vector de características.

Es relevante mencionar que se debe tener especial cuidado con no realizar una cantidad excesiva de consultas a un mismo sitio web y entregar un *user/agent* normal (Navegador web) pues el sitio detectará con facilidad que las consultas son realizadas por un programa (bot) y se denegará el acceso y por consiguiente no será posible rescatar información[55].

3.4.2. Certificado de seguridad

La métrica utilizada para medir seguridad serán los certificados de seguridad SSL (secure sockets layer) o capa de puertos seguros, los cuales son protocolos criptográficos utilizados para proteger sistemas contra ataques impersonales y técnicas de *eavesdropping* (escuchar información secretamente)[56], en este caso particular entre un usuario y un sitio web.

El certificado de seguridad se obtiene a través de un servicio web, el cual lo provee SSL LABS, quienes además disponen de una librería para Java para la utilización de su servicio web.

Capítulo 4

Implementación del vector de características

En este capitulo se detallará el diseño, desarrollo e implementación del vector de características. El objetivo de este capítulo es recolectar información con respecto a características diferenciadoras entre los distintos tipos de páginas web, luego se detalla el proceso de obtención de variables relacionadas con contenido HTML y seguridad web, y finalmente se describirá el proceso de desarrollo e implementación del vector de características en un módulo de Java.

Con el objetivo de analizar las características de las distintas clases de páginas web, es necesario construir el juego de datos previamente etiquetado y de esta forma analizar en una primera etapa, visualmente, las similitudes entre las observaciones dentro de una misma clase y las características excluyentes entre una clase y otra.

4.1. Construcción de juego de datos

La construcción del juego de datos no consiste simplemente en la recolección de datos al azar. Con el fin de obtener valor a partir de los datos la construcción del juego de datos debe venir acompañado del contexto (como se capturan), como son procesados, analizados y validados[57].

De esta manera se construye el juego de datos a partir de páginas web pertenecientes a los sitios web más visitados por usuarios Chilenos[1] de los cuales se filtran ciertos sitios web que podrían aportar ruido excesivo a la muestra, como por ejemplo sitios de muy difícil clasificación para el usuario, contenido ilegal y sitios web orientales, el último tiene gran relevancia puesto que la barrera de idioma si bien es un problema al momento de clasificar, este puede ser abordado bajo distintos enfoques de procesamiento de texto sin comprometer demasiado sus resultados. El problema real es que los sitios web orientales difieren de los occidentales en contenido y diseño a nivel cultural, son visualmente más complejos, más información, contenido interactivo e incluso mayor seguridad [58]. Para la tarea encomendada se utilizan los resultados que ofrece Alexa[2], plataforma de marketing digital capaz de ordenar sitios web con respecto a su tráfico por país, para construir el juego de

[1] El juego de datos se construye en su mayoría por los sitios más visitados por usuarios Chilenos pero con el objetivo de generar una muestra más representativa, se utilizan también sitios muy visitados por usuarios Argentinos y Peruanos

[2] http://www.alexa.com/topsites/countries/CL

datos.

Captura de muestras para el juego de datos

El proceso de selección de observaciones para el juego de datos se realiza en un archivo de valores separados por coma (CSV) desde la plataforma Alexa. Considerando que existen siete posibles clases para clasificar una página web, se define que se necesitan 150 observaciones para preparar el set de entrenamiento y set de testeo.
Luego de terminada la captura de observaciones, se procede al proceso de etiquetado.

Tabla 4.1: Captura Set URL

1	URL
2	https://www.youtube.com
3	https://www.google.cl
4	https://www.facebook.com
5	http://www.emol.com
...	...
151	http://www.udd.cl

4.1.1. Etiquetado de juego de datos

El proceso de etiquetado de las observaciones consiste en asignar manualmente a cada observación (URL), una categoría, es decir clasificar cada página web con respecto a su tipo de pagina web (Motor de búsqueda, Servicios financieros, informativa, entretenimiento, noticias, E-commerce, red social). Este proceso es necesario puesto que la definición del problema requiere categorías predefinidas, lo que motiva la utilización de algoritmos de aprendizaje supervisado.

Es recomendable al menos en el set de entrenamiento utilizar URLs que pertenezcan a solo una categoría predominante, de esta manera es más fácil encontrar variables y patrones excluyentes entre categorías, es por esta razón que al menos en esta etapa se eliminan registros de URL demasiado complejas de analizar.

Tabla 4.2: Etiquetado Set URL

1	URL	Label
2	https://www.youtube.com	Entretenimiento
3	https://www.google.cl	Motor de búsqueda
4	https://www.facebook.com	Red social
5	http://www.emol.com	Noticias
...	...	...
151	http://www.udd.cl	Información

4.2. Características de sitios web por categoría

Una página web es diferenciable con respecto al servicio que este ofrece de tal manera que no puede pertenecer a dos categorías, si bien una página web puede tener rasgos de más de una categoría, siempre va a poder ser clasificada en la categoría predominante. La labor de clasificar una página web se realiza de forma muy sencilla por los usuarios, generalmente en el primer minuto dentro de un sitio un usuario es capaz de reconocer cual es el objetivo del sitio. Esta labor se intenta replicar en el presente trabajo de título para lo cual se realiza una diferenciación en base a diseño y en base a contenido HTML.

4.2.1. Características según diseño

En esta sección se pretende diferenciar una página web de otra con respecto al contenido visual, solo se considera el diseño que es apreciable por el usuario que navega la página principal de un sitio web.

Este análisis es una elaboración propia puesto que se pretende realizar una clasificación personalizada para la realidad de las páginas web latinoamericanas con fuerte énfasis en las páginas web Chilenas.

4.2.1.1. Motores de búsqueda

Los motores de búsqueda son fáciles de identificar para el usuario, generalmente tienen un diseño simple y limpio (pocos objetos) en donde predomina la caja de texto que permite realizar la búsqueda.
El reconocimiento de un motor de búsqueda es particularmente complejo en páginas web híbridas cuya función principal es la de ser un motor de búsqueda, puesto que la mayor parte del contenido visual no pertenece a su función objetivo (ejemplo: `www.yahoo.com`).

1. *Seach box* de gran tamaño.
2. Palabras clave: “buscar” o similares.
3. Presencia de *log in*.
4. Acceso a correo.

Las dos últimas características son propias de motores de búsqueda reconocidos y de gran tamaño. No aplica a buscadores independientes.

4.2.1.2. E-Commerce

La identificación de páginas web relacionadas con E-commerce es particularmente difícil puesto que estas páginas web tienen una alta variabilidad entre si tanto en estructura visual como HTML. A pesar de lo anterior se pueden reconocer características comunes si exceptuamos las páginas web pertenecientes a E-commerce Business to Business, las cuales serán omitidas en este apartado.

1. *Search box* de tamaño variable.
2. Presencia de login.
3. Existencia de *carro de compra*.
4. Palabras clave: “$”, “Dcto”,“Descuento”,“Compra”,“Venta”, “Métodos de pago”, “Medios de pago”, ”cambios”,”Devoluciones”, ”como comprar”.

Las páginas web pertenecientes a E-commerce B2B no serán clasificadas bajo la categoría E-commerce, puesto que en estructura es suficientemente distinta y no pretende vender a través del sitio web directamente. Una página web B2B pretende incitar al usuario a informarse y contactar a la empresa ofertante, teniendo en cuenta esto, su clasificación final para los fines expuestos será informativa.

4.2.1.3. Entretenimiento

Las páginas web pertenecientes a esta clase poseen una gran cantidad de contenido audiovisual y con estructura muy variada, además tratan tópicos de diversos índoles, lo que hace fundamental la identificación de variables excluyentes, dado que identificar la página web en su globalidad es una tarea muy compleja.

1. Alto contenido de imágenes y vídeos.
2. *Search box* de tamaño variable.
3. Palabras clave: “música”, “deportes”, “televisión”, “arte”, “videojuegos”.

4.2.1.4. Noticias

Páginas web de noticias se destacan por una gran saturación del espacio entre imágenes y texto con relación a actualidad. Es importante destacar que es la única clase que le entrega importancia a tener algún indicador temporal (fecha, día, hora) y como es de esperar los tópicos pertenecen a alguna sección de los diarios físicos, las cuales son comunes para la mayoría de los diarios online.

1. Indicador Temporal.
2. Alto contenido de texto e imágenes.

3. *Search box* pequeño.

4. Login.

5. Palabras clave: “Noticias”, “Economía”, “Deportes”, “Expectáculos”, “Tendencias”, “Servicios”, “Fotos”, “Avisos”, “Cultura”, “Mundo”, “Nacion”, “Nacional”, “Internacional”, “Negocios”.

4.2.1.5. Información

Existen dos tipos predominantes de páginas web informativas, páginas cuyo objetivo es educar, en las cuales predomina el texto plano (ejemplo: páginas wiki), la segunda categoría son páginas web explicativas sobre algún tópico, labor o producto, ejemplo de esto son las páginas de *kickstarter*, que en contraste tienen imágenes de gran tamaño, texto medio y contenido dinámico (vídeos o cintas ”carousel”).

1. Alto contenido de texto.

2. Pequeño *Search box*.

3. Contenido educativo.

4.2.1.6. Servicios Financieros

Las páginas web relacionadas con servicios financieros se enfocan en el usuario, el cual generalmente es el cliente, es por esto que es fundamental que el login sea muy visible, además es habitual que ofrezcan beneficios en una cinta ”carousel”. Finalmente todas estas instituciones tienen un pequeño apartado citando a la superintendencia de bancos e instituciones financieras.

1. Banner estilo carousel.

2. *Search Box* pequeño.

3. Presencia de login.

4. Servicio al cliente.

5. Alto contenido de imágenes.

6. Palabras clave: “banco”, “personas”, “empresas”, “indicadores económicos”, “PYME”, “joven”, “personas”.

4.2.1.7. Red Social

Las páginas clasificadas como redes sociales al igual que los motores de búsqueda poseen un diseño simple y limpio, el cual a diferencia de los segundos ponen énfasis en un login y reclutamiento de miembros, ya sea gratis o de pago. Es común que en esta categoría se explique el beneficio de la red social y una imagen de gran tamaño se utiliza como fondo.
Considerar que los sitios de citas online se consideran dentro de esta categoría.

1. Login de gran tamaño.
2. Formulario de registro.
3. Hipervínculos a usuarios.
4. Palabras clave: “noticias”, “amigos”, “seguidores”, “”,“#”.

4.2.2. Desarrollo de Software y Características según contenido HTML

El desarrollo de software se realiza en Netbeans IDE bajo el lenguaje Java y la librería JSOUP. El algoritmo del software consiste en la carga de URLs previamente clasificadas en un arreglo, se inicia un contador para la lectura de registros, si existe el registro que apunta el contador se lee la URL y se conecta al servidor correspondiente, se extrae el contenido HTML y se realiza el minado de características (Web Crawling). Luego de llenado el vector de características para el primer registro (URL), se escribe la URL, etiqueta (clase de la URL) y variables del vector de características en un archivo CSV (”output.csv”). El proceso se repite hasta que no existan registros en la base de datos de URL.

Se deben tener ciertas consideraciones al realizar un Web Crawling de estas características:

1. ***User Agent***

   Es necesario declarar el “User Agent”, el cual en este caso sera: “Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_4) AppleWebKit/601.5.17 (KHTML, like Gecko) Version/9.1 Safari/601.5.17”. Esto se realiza con el objetivo de mostrarle al servidor web un browser válido y este no deniegue acceso por peticiones poco comunes (no son enviadas utilizando el protocolo HTTP), es decir que el servidor web no asuma que las peticiones son enviadas desde un bot.

   Es importante destacar que no se deben enviar demasiadas peticiones seguidas al mismo servidor web (varias páginas pertenecientes al mismo sitio web), puesto que también se denegará el acceso por la misma razón expuesta anteriormente.

   Este problema se soluciona con relativa facilidad, por un lado se intenta no realizar extracción de contenido de dos páginas pertenecientes a un mismo sitio web de manera contigua y en caso de realizarlo se asigna un periodo de reposo, el cual debe ser suficiente para no exceder

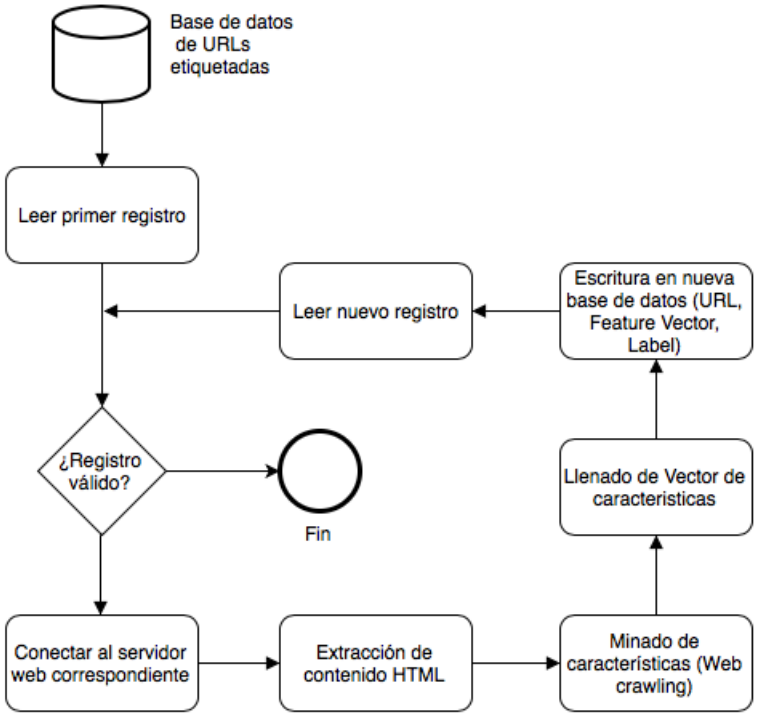


Figura 4.1: Flujo de software

las 60 peticiones (en este caso son dos[3] peticiones por página web), en el caso de este trabajo no es necesaria la implementación de esta medida, sin embargo para el producto final podría ser relevante este análisis.

2. ***Time out***

   El time out es el tiempo de espera definido para encontrar una respuesta y luego de transcurrido ese tiempo seguir a la próxima observación, en este caso particular se declara tiempo infinito puesto que todas las URL fueron escogidas manualmente y son URL funcionales y válidas, considerar que el llenado del vector de características puede variar drásticamente entre una URL y otra.

3. ***Language***

   El idioma se intenta forzar a español si existe en la página, esto con el objetivo de poder realizar análisis de texto, sin embargo de no poder ser así, la clasificación se lleva a cabo de todas formas.

Tomando en consideración la caracterización según diseño comienza la construcción del vector de características.

1. ***INPUT SUBMIT***

   Esta es una variable de envió de formularios, su justificación es la identificación de "lo-

[3]Una petición para rescatar contenido HTML y otra petición para rescatar el certificado de seguridad

gin" y *Search box*. En este caso particular solo es de interés saber si existe, no es importante si existen múltiples campos "enviables", por lo que corresponde a una variable binaria.

```
Data: Listado de URLs
Result: Variable binaria INPUT_SUBMIT
initialization i=1;
while i<n do
    foreach URL ∈ URLDataBase do
        Write URL;
        if URL[i].Select("input[type = submit])=!null) then
            INPUT_SUBMIT=1;
        else
            INPUT_SUBMIT=0;
        end
    end
    Write INPUT_SUBMIT;
    Next i;
end
```

Algorithm 4.1: Implementación de variable INPUT SUBMIT

2. ***PASSWORD***

La variable anterior es incapaz de identificar un login por si sola, puesto que existen múltiples objetos que requieren que el usuario envié información a los servidores, como por ejemplo utilizar un buscador únicamente a través de su *Text box*. La variable PASSWORD identifica la mayoría de los posibles nombres que tiene el objeto que designa al campo que requiere al usuario completar con una contraseña, se espera que ese objeto sea parte de un formulario de login.
La variable corresponde a una variable binaria.

```
Data: Listado de URLs
Result: Variable binaria PASSWORD
initialization i = 1;
while i<n do
    foreach URL ∈ URLDataBase do
        Write URL[i];
        if URL[i].Select("input[type = password]") =! null) then
            PASSWORD = 1;
        else
            PASSWORD = 0;
        end
    end
    WRITE PASSWORD;
    Next i;
end
```

Algorithm 4.2: Implementación de variable PASSWORD

3. ***IMAGE***

La variable IMAGE es una variable de conteo, que cuenta cuantos objetos en la página web corresponden a imágenes estáticas, esta variable pretende ayudar a la distinción de páginas web de entretenimiento, E-commerce y Noticias.

```
Data: Listado de URLs
Result: Variable entera IMAGE
initialization i = 1;
while i<n do
    foreach URL ∈ URLDataBase do
        Write URL[i];
        if URL[i].Select("img").attr("src =") =! null) then
            IMAGE = # de apariciones de scr;
        else
            IMAGES = 0;
        end
        if URL[i].Select("img").attr("map =") =! null) then
            IMAGE = IMAGE + # de apariciones de map;
        else
            IMAGES = IMAGES + 0;
        end
        if URL[i].Select("img").attr("area =") =! null) then
            PASSWORD=IMAGE + # de apariciones de area;
        else
            IMAGES = IMAGES + 0;
        end
    end
    Write IMAGE;
    Next i;
end
```

Algorithm 4.3: Implementación de variable IMAGE

4. ***BUTTON SUBMIT***

La presente variable se diferencia de la primera en la necesidad de enviar un formulario a través de un botón, lo que implica una gran diferencia en estructura, no necesariamente un formulario de tipo login posee un botón de tipo *submit*, pero siempre este botón esta presente en todo formulario de búsqueda.

La variable descrita es una variable complementaria que ayuda a levantar información y realizar diferencias entre tipos de páginas web. La variable BUTTON SUBMIT es una variable binaria.

```
Data: Listado de URLs
Result: Variable binaria BUTTON_SUBMIT
initialization i = 1;
while i<n do
    foreach URL ∈ URLDataBase do
        Write URL[i];
        if URL[i].Select("button[type = submit]") =! null) then
            BUTTON_SUBMIT = 1;
        else
            BUTTON_SUBMIT = 0;
        end
    end
    WRITE BUTTON_SUBMIT;
    Next i;
end
```

Algorithm 4.4: Implementación de variable BUTTON SUBMIT

5. ***GET TIME***

La variable GET TIME nace de un análisis exploratorio en el cual se concluye que existen solo dos tipos de páginas que requieren la obtención de métricas temporales (fecha, día, hora), las cuales son páginas web de noticias y servicios financieros, si bien pueden existir excepciones, no es la regla.
La variable corresponde a una de tipo binaria.

```
Data: Listado de URLs
Result: Variable binaria GET_TIME
initialization i = 1;
while i<n do
    foreach URL ∈ URLDataBase do
        Write URL[i];
        if URL[i].Select("script[type = text/javascript]").text.contains("getTime()") =!
          null) then
            GET_TIME = 1;
        else
            GET_TIME = 0;
        end
    end
    WRITE GET_TIME;
    Next i;
end
```

Algorithm 4.5: Implementación de variable GET TIME

6. ***CARRO COMPRAS***

El carro de compras es un objeto muy característico y excluyente de los sitios web de retail, los cuales son un subconjunto que representa la mayor parte de los E-commerce actuales, es por eso que es de gran relevancia la creación de una variable binaria que sea capaz de identificar la presencia de dicho objeto. La variable que apunta al objeto de carro de compras es una variable binaria.

```
Data: Listado de URLs
Result: Variable binaria CARRO_COMPRAS
initialization i = 1;
while i<n do
    foreach URL ∈ URLDataBase do
        Write URL[i];
        if URL[i].Select("svg").text.contains("compras||carro||cart||bolsa") =! null) then
            CARRO_COMPRAS = 1;
        else
            CARRO_COMPRAS = 0;
            if URL[i].Select("div").text.contains("compras||carro||cart||bolsa") =! null)
              then
                CARRO_COMPRAS = 1;
            else
                CARRO_COMPRAS = 0;
            end
        end
    end
    WRITE CARRO_COMPRAS;
    Next i;
end
```

Algorithm 4.6: Implementación de variable CARRO COMPRAS

7. ***FAQ***

El FAQ o frequently asked questions (preguntas frecuentes), es una sección propia de sitios web en los cuales se espera que el cliente pueda tener problemas con cierto contenido que ofrece el sitio, ya sea con una acción particular o con el objetivo de algún módulo de este. Un sitio web que posee FAQ tiene cierto grado de complejidad y predomina en sitios web de E-commerce, Servicios financieros e Informativos, en donde hay servicios web e información más compleja.
La variable relacionada con contenido FAQ es una variable binaria.

```
Data: Listado de URLs
Result: Variable binaria FAQ
initialization i = 1;
while i<n do
    foreach URL ∈ URLDataBase do
        Write URL[i];
        if URL[i].Select("a[href]").text.contains("faq") =! null) then
            FAQ = 1;
        else
            FAQ = 0;
        end
    end
    WRITE FAQ;
    Next i;
end
```

Algorithm 4.7: Implementación de variable FAQ

8. ***PRICE TAG***

Entendemos por *price tag* todo lo relacionado con la etiqueta que muestra el valor de un producto, en el caso de las páginas web generalmente esta relacionado con caracteres numéricos que siguen al signo monetario "$". El símbolo $ es muy característico de sitios web de E-commerce, pero no es exclusivo de estos, pues páginas de muchos tópicos pueden hablar de dinero, lo que lo hace una variable fuerte pero no decisiva.
La variable descrita corresponde a una variable binaria.

```
Data: Listado de URLs
Result: Variable binaria PRICE_TAG
initialization i = 1;
while i<n do
    foreach URL ∈ URLDataBase do
        Write URL[i];
        if URL[i].Select("span").text.contains("$") =! null) then
            PRICE_TAG = 1;
        else
            PRICE_TAG = 0;
        end
    end
    WRITE PRICE_TAG;
    Next i;
end
```

Algorithm 4.8: Implementación de variable PRICE TAG

9. ***COPYRIGHT***

El Copyright nace como una herramienta para la protección de derechos de autor, por lo tanto esto solo tiene sentido cuando el dueño de un sitio web pretende proteger una marca, por consiguiente el único tipo de páginas web que no siempre se considera son las páginas de entretenimiento.

```
Data: Listado de URLs
Result: Variable binaria COPYRIGHT
initialization i=1;
while i<n do
    foreach URL ∈ URLDataBase do
        Write URL[i];
        if URL[i].Select("div").text.contains("derechosreservados") =! null) then
            COPYRIGHT = 1;
        else
            COPYRIGHT = 0;
        end
    end
    WRITE COPYRIGHT;
    Next i;
end
```

Algorithm 4.9: Implementación de variable COPYRIGHT

10. ***EMAIL LINK***

La variable EMAIL LINK hace referencia a la existencia de un hipervínculo u objeto que tenga por finalidad apuntar a un formulario de correo o simplemente hacer referencia al contacto del sitio web. Si bien pareciera común que los dueños o desarrolladores habrán la posibilidad de dialogo con los usuarios, no en todos los tipos de sitios web destinan un espacio para realizar dicha tarea. Los sitios web con mayor probabilidad de tener esta característica son E-commerce, informativos y servicios financieros.
Debido a sus características la variable EMAIL LINK es una variable binaria.

```
Data: Listado de URLs
Result: Variable binaria EMAIL_LINK
initialization i = 1;
while i<n do
    foreach URL ∈ URLDataBase do
        Write URL[i];
        if URL[i].Select("a").text.contains("mailto") =! null) then
            EMAIL_LINK = 1;
        else
            EMAIL_LINK = 0;
        end
    end
    WRITE EMAIL_LINK;
    Next i;
end
```

Algorithm 4.10: Implementación de variable EMAIL LINKS

11. ***DESCRIPTION***

La variable DESCRIPTION es una variable polinominal, la cual está construida por contenido HTML no visual para el usuario, esta variable recolecta información que provee el programador de la página acerca de esta misma y busca palabras clave que podrían dar indicios de a que tipo de página web pertenece.
A continuación se describen las bolsas de palabras que pretenden caracterizar a cada sitio web bajo la variable relativa a la descripción.

(a) E-commerce:
Palabras clave: ”retail”, ”tienda”, ”compra”.

(b) Servicios financieros:
Palabras clave: "banco", "bancario", "financiero", "finanzas", ”crédito", ”inversión", "pyme";

(c) Motor de búsqueda
Palabras clave: ”busqueda”, ”buscar”, ”search”, ”engine”, ”motor”.

(d) Noticias
Palabras clave: ”noticias”, ”informate”, ”informacion”, ”informar”, ”acontecer”, ”diario”, ”medio”.

(e) Entretenimiento
Palabras clave: ”farandula”, ”entretenimiento”, ”musica”, ”peliculas”, ”television”, ”celebridades”, ”juegos”, ”videos”.

(f) Red social
Palabras clave: ”red”, ”social”, ”conectar”, ”conectate”, ”amigos”, ”conecta”, ”comparte”,”citas”,”amor”.

(g) Información
Palabras clave: ”colegio”, ”universidad”, ”instituto”, ”formacion”, ”aprender”, ”gobierno”, ”aprende”.

```
Data: Listado de URLs
Result: Variable binaria DESCRIPTION
initialization i = 1;
while i<n do
    foreach URL ∈ URLDataBase do
        Write URL[i];
        if URL[i].Select("meta[name =
          description]").text.contains("retail"|"tienda"|"compra")=!null) then
            E_COMMERCE=1;
        else
            E_COMMERCE=0;
        end
        if URL[i].Select("meta[name =
          description]").text.contains("financiero"|"banco"|"bancario"|
        "finanzas"|"credito"|"inversion"|"pyme") =! null) then
            SERVICIOS_FINANCIEROS = 1
        else
            SERVICIOS_FINANCIEROS = 0
        end
        if URL[i].Select("meta[name =
          description]").text.contains("busqueda"|"buscar"|"search"|"engine"|"motor") =!
          null) then
            MOTOR_BUSQUEDA = 1
        else
            MOTOR_BUSQUEDA = 0
        end
        if URL[i].Select("meta[name =
          description]").text.contains("noticias"|"informate"|"informacion"|"informar"
        |"acontecer"|"diario"|"medio") =! null) then
            NOTICIAS = 1
        else
            NOTICIAS = 0
        end
        if URL[i].Select("meta[name =
          description]").text.contains("farandula"|"entretenimiento"|"muscia"|"peliculas"
        |"television"|"celebridades"|"juegos"|"videos") =! null) then
            ENTRETENIMIENTO = 1
        else
            ENTRETENIMIENTO = 0
        end
        if URL[i].Select("meta[name =
          description]").text.contains("red"|"social"|"conectar"|"conectate"|"amigos"
        |"conecta"|"comparte"|"citas"|"amor") =! null) then
            RED_SOCIAL= 1
        else
            RED_SOCIAL = 0
        end
        if URL[i].Select("meta[name =
          description]").text.contains("colegio"|"universidad"|"instituto"|"formacion"
        |"aprender"|"gobierno"|"aprende") =! null) then
            INFORMACION = 1
        else
            INFORMACION = 0
        end
    end
    WRITE E_COMMERCE;
    WRITE SERVICIOS_FINANCIEROS;
    WRITE MOTOR_BUSQUEDA;
    WRITE NOTICIAS;
    WRITE ENTRETENIMIENTO;
    WRITE RED_SOCIAL;
    WRITE INFORMACION;
    Next i;
end
```

Algorithm 4.11: Implementación de variable DESCRIPCION

12. ***PERSONAS|EMPRESAS***

La variable PERSONAS|EMPRESAS pretende identificar los paneles propios de los sitios de servicios financieros que apuntan tanto a formularios de ingreso para “personas” y para “empresas”.
La variable recién descrita corresponde a una variable de tipo binaria.

```
Data: Listado de URLs
Result: Variable binaria PERSONAS|EMPRESAS
initialization i = 1;
while i<n do
    foreach URL ∈ URLDataBase do
        Write URL[i];
        if URL[i].Select("a").text.contains("a").text.contains("personas"|"empresas") =!
         null) then
            PERSONAS|EMPRESAS = 1;
        else
            PERSONAS|EMPRESAS = 0;
        end
    end
    WRITE PERSONAS|EMPRESAS;
    Next i;
end
```

Algorithm 4.12: Implementación de variable PERSONAS|EMPRESAS

13. ***SUPERINTENDENCIA DE BANCOS Y SERVICIOS FINANCIEROS***

La variable SUPERINTENDENCIA DE BANCOS Y SERVICIOS FINANCIEROS verifica la existencia de contenido relacionado con la garantía estatal que entrega la SBIF hacia bancos y servicios financieros en general, lo cual siempre se encuentra explícitamente en sitios web de servicios financieros.
La variable corresponde a una de tipo binaria.

```
Data: Listado de URLs
Result: Variable binaria SBIF
initialization i = 1;
while i<n do
    foreach URL ∈ URLDataBase do
        Write URL[i];
        if URL[i].Select("p").text.contains("sbif"|"www.sbif.cl") =! null) then
            SBIF = 1;
        else
            SBIF = 0;
        end
    end
    WRITE SBIF;
    Next i;
end
```

Algorithm 4.13: Implementación de variable SBIF

14. ***CINTA DINÁMICA***

La presente variable identifica contenido visual móvil, es decir imágenes no estáticas, ya sea una cinta de imágenes de tipo carousel, activada por timer o simplemente imágenes rotacionales activadas por usuario (click). Corresponde a una variable de tipo binaria.

```
Data: Listado de URLs
Result: Variable binaria CINTA_DINMICA
initialization i = 1;
while i<n do
    foreach URL ∈ URLDataBase do
        Write URL[i];
        if URL[i].Select("div").attr(class).text.contains("timer"|"self − activate −
          slide"|"itemactive")=!null) then
            CINTA_DINMICA = 1;
        else
            CINTA_DINMICA = 0;
        end
    end
    WRITE CINTA_DINMICA;
    Next i;
end
```

Algorithm 4.14: Implementación de variable CINTA DINÁMICA

15. ***DOMINIO***

La variable dominio se refiere a una variable nominal que rescata el nombre del dominio (DOMAIN NAME) de un sitio web (extrapolable a páginas web). El dominio es una variable muy importante, ya que ciertos dominios son exclusivos de algunas clases de páginas web, por ejemplo el dominio *.tv* es exclusivo de sitios web de entretenimiento.
Notar que se obtienen las tres primeras letras luego del segundo punto de la URL, lo cual no necesariamente es el dominio, ya que si el dominio tiene solo dos letras (ejemplo: *cl*) y además posee *ruta*, el presente algoritmo recuperará el dominio, seguido del carácter "/" (ejemplo: *cl/*), lo cual es una variable nominal distintiva.

Data: Listado de URLs
Result: Variable binaria *DOMINIO*
initialization i=1;
while $i<n$ **do**
  foreach $URL \in URLDataBas$e **do**
    DOMINIO = SubString(Second("."),3);
  end
  WRITE *DOMINIO*;
  Next i;
end

Algorithm 4.15: Implementación de variable DOMINIO

4.2.3. Seguridad según SSL/TLS

La razón detrás de la aplicación de seguridad web como variable predicativa descansa sobre la creencia que distintas clases de sitios web requieren distintos niveles de seguridad[45], por lo tanto esta variable debiese entregar información valiosa para complementar el estudio anterior. Además es importante destacar que si bien la seguridad que provee el protocolo HTTPS es dudosa[59], un certificado de seguridad bien configurado junto a dicho protocolo provee la seguridad web sugerida para la mayoría de las entidades en la web, este estudio sugiere de cierta manera que un estudio de los protocolos de seguridad no sería suficiente para clasificar páginas web si la seguridad web es un factor real para distinguir clases.

Certificado X.509

Un certificado de tipo X.509 es en criptografía el estándar de PKI (Public Key Infraestructure), utilizado para manejar certificados digitales y "Public Key Encryptions"[60], lo cual a su vez es parte fundamental del protocolo TSL (se detalla en el capítulo 2).

La construcción del Set de variables relativo a seguridad con respecto a SSL/TSL se desea construir a partir de la extracción de certificados de tipo X.509, sin embargo la primera pregunta que debemos hacernos es si efectivamente la página web posee certificados, en el caso particular de este trabajo, el estudio presenta cierto sesgo, pues en instancia puede suceder que sitios web compran certificados para ciertas páginas web y no para la totalidad de su sitio web, en particular para páginas web cerca del final del embudo de conversión (ejemplo: carro de compra), por lo tanto se

definen como páginas sin protección SSL si no poseen el certificado para la URL dada, pudiendo tener dicho certificado en una URL distinta perteneciente al mismo sitio web.

4.2.3.1. SSL Server Rating

El análisis de certificados de seguridad se desea realizar a través de los servicios que ofrece SSL LABS[4], puesto que no solo provee las características de los certificados de seguridad, si no que las analiza, interpreta y mediante ponderadores entrega una categorización de sitios web según seguridad a través de un sistema de rating.

El sistema de rating de SSL LABS es un proceso de asignación de puntaje a sitios web a través de una suma ponderada de tres factores (ver metodología para asignación de puntaje según SSL LABS en capítulo 2), el cual resulta en una letra entre A y F según el puntaje obtenido.

4.2.3.2. Inclusión de Variable de Seguridad web al juego de datos

La inclusión de la variable relativa a seguridad web no es un proceso directo, puesto que lo que se requiere no es simplemente el certificado de seguridad, si no que es el puntaje que otorga una entidad a los distintos sitios web, para resolver este problema se visualizan dos soluciones.

1. ***Web Scrapper:*** La primera solución consiste en modificar el módulo que se utilizó en la etapa de la creación del vector de características según contenido web, el cual recolectaba objetos específicos dentro de una página web. Es posible realizar un trabajo similar en el sitio web de SSLLABS, sobre su aplicación web que realiza la labor de asignar puntaje [5] y luego realizar iteraciones para cada página web del juego de datos y rescatar la letra asignada (puntaje).

2. ***SSLLABS API:*** Afortunadamente colaboradores externos han desarrollado dos API públicas para Java, desafortunadamente dicha aplicación no fue diseñada para los requerimientos de este trabajo y abarcan un marco más extenso, lo que implica modificaciones y entendimiento del trabajo de otros, lo cual puede ser suficiente razón para inclinarse por la primera opción.

La primera opción se descarta debido a que el proceso de asignar puntaje a cada URL es un proceso que consume bastante tiempo y dependiendo del sitio web no siempre se redirecciona directamente a la página donde se muestra el puntaje, en algunos caso puede redireccionar a una página web de error o a una página web que muestra todos los servidores relacionados al servidor que se quiere evaluar. Esta opción se descarta solo si la segunda opción presenta menores problemas.

La segunda opción presenta una implementación bastante más compleja que como se había estimado, no existe un módulo que permita conectarse a la API y entregue resultados inmediatos, sin embargo dentro de la aplicación existe una aplicación web que dentro de sus resultados se muestra la nota otorgada por el servidor web, es decir muestra el último resultado que el sitio web

[4]https://www.ssllabs.com

[5]Disponible en `https://www.ssllabs.com/ssltest/`

de SSL Labs le ha otorgado al sitio web en cuestión, estos no son resultados en tiempo real, es una vista de la base de datos de SSL LABS, si el resultado no se encuentra en la base de datos, resultará en el mismo resultado de un sitio web sin certificado SSL.

Se decide ir por una tercera opción que implica una mezcla de las anteriores haciendo énfasis en la aplicación web, se realiza un Web Scrapper sobre los resultados de aplicación web de la API, los cuales se expresan en formato *JSON* (*JavaScript Object Notation*), por lo que es necesario realizar una transformación a texto y luego agregarlo al antiguo juego de datos (vector de características). Se encontraron algunos problemas de implementación al momento de realizar la iteración para todas las URL los cuales se exponen a continuación:

1. Se presenta en reiteradas ocasiones el error 429[6], lo cual se resuelve entregando un User-Agent válido (Mozilla 5.0), junto con limitar analizar una URL cada dos segundos.

2. Las páginas web cuyo certificado no puede ser encontrado se analizan manualmente en el sitio web de SSL LABS, en el caso de que realmente el certificado no se encuentra implementado, el proceso es muy rápido, el caso contrario toma tiempo y se agrega a la base de datos para ser encontrado en la nueva iteración.

La implementación en su forma final (luego de hacerse cargo de las consideraciones anteriores) consiste en conectarse a la API en su versión web, reconocer el objeto en donde se encuentra la nota (en este caso se llama ”endpoints”), rescatar la nota y escribirla en un archivo *CSV*, en caso de no encontrar nota se escribe *"S/C"* (sin certificado).

Data: Listado de URLs
Result: URL, Puntaje (Letras)
initialization i=1;
while *i<n* **do**
  read URL[i];
  Write URL[i];
  if *(connect to: “https://api.ssllabs.com/api/v2/analyze?host=” + URL[i] == SUCCESSFUL AND get(endpoints).get(grade) != null)* **then**
    Write *grade*;
  else
    Write S/C as *grade*;
  end
  Next i;
end

Algorithm 4.16: Implementación de variable de seguridad SSL

El siguiente paso corresponde a una transformación en el sistema de puntaje, existen seis notas básicas (A,B,C,D,E,F) y dos notas especiales $(A+,A-)$, por lo tanto para evitar complejizar el modelo sin un beneficio claro, se agrupan $(A-,A,A+)$ dentro de la misma nota (A) (figura 4.2) . Finalmente se procede a agregar dicha variable al vector de características (ver figura **??**).

[6]Error 429: El usuario ha realizado demasiadas solicitudes en una ventana de tiempo dada.

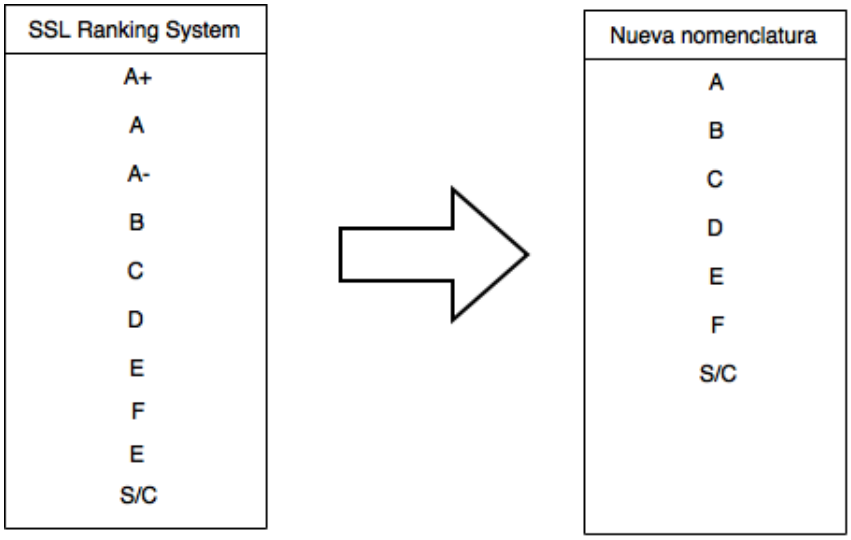


Figura 4.2: Transformación de nomenclatura de SSL Ranking System
Fuente: Elaboración propia

Capítulo 5

Minería de datos

Una vez construida la base de datos que contiene las columnas [URL, Label, Vector de características], comienza el proceso de minería de datos. Para la minería de datos y en particular para el proceso de extracción de conocimiento a partir de datos se utilizará la metodología KDD (Knowledge Discovery in Databases), la cual provee pasos a seguir de manera de estandarizar el proceso con buenos resultados.

5.1. Especificaciones técnicas

El proceso de minería de datos puede ser un proceso extremadamente computacionalmente exhaustivo, por lo tanto el tiempo de resolución de un problema e incluso la factibilidad de esta solución tiene relación directa tanto con el software como con el hardware a utilizar. Con el objetivo de entregar una correcta documentación del proceso de toma de datos y obtención de resultados (además de hacer posible una replicación del experimento), se intentará detallar de mejor manera todo material que tuve algún impacto en la elaboración del experimento.

5.1.1. Hardware

En lo que respecta a Hardware solo se utilizará un computador portátil tanto para la toma de datos como para el procesamiento de estos, el cual se describe a continuación:

MacBook Pro (13-inch, Mid 2012) Procesador: 2,5 GHz Intel Core i5 Memoria: 4 GB 1600 MHz DDR3 Gráficos: Intel HD Graphics 4000 1536 MB

5.1.2. Software

Se detallará no solo el software utilizado para el proceso de minería de datos si no que también todo el software utilizado para el proceso de creación del juego de datos.

Juego de datos:

Sistema Operativo X El Capitán versión 10.11.4
Entorno de desarrollo: NetBeansIDE 8.1
Navegador web: Mozilla 5.0

Minería de datos:

Sistema Operativo X El Capitán versión 10.11.4
Modelamiento: Rapidminer Studio 7.1.001

5.2. Modelamiento

El proceso de modelamiento es un proceso iterativo, en este trabajo de titulo se pretende detallar el proceso que llega al resultado final con el objetivo de no solo mostrar resultados si no que también todos los caminos tomados con respecto a los cuales se pueden realizar recomendaciones de que practicas repetir y cuales evitar.

Tal como se define en el capitulo anterior, se define un juego de datos de 150 observaciones, sobre los cuales se realiza una validación cruzada de 10 iteraciones, por lo que se define un set de entrenamiento de 90% y testeo 10%. Se utiliza Validación cruzada puesto que esta técnica garantiza resultados independiente de la partición lo que es muy importante para evitar sobreajuste de los datos y obtener resultados más representativos.

Como medida de desempeño se analizan las dos medidas más utilizadas, por un lado *accuracy* nos entrega el porcentaje de observaciones (en este caso páginas web) clasificadas correctamente y por otro lado *F-measure* o *F-score* corresponde a la media armónica entre *precision* y *recall*, que debido a que es una métrica mixta no tiene una interpretación intuitiva, si no existe prioridad entre *precision* y *recall*, entrega un valor que refleja el desempeño conjunto de dichas métricas. Notar que si bien es una fracción cuyo mejor valor es 1 y su peor valor es 0, no puede ser interpretado como un porcentaje, puesto que por construcción, no lo es.

F-measure es excepcionalmente bueno para experimentos en los que existen clases no balanceadas como es el caso de este trabajo, sin embargo, debido a que el juego de datos se construyó a partir de *rankings* según tráfico web, se pretende asignar mayor importancia a las clases que tienen mayor cantidad de observaciones y por consiguiente mayor interés del mercado.

Debido a que el proceso de construcción del juego de datos se realizó de forma manual no se realiza limpieza con respecto a formato ni con respecto a valores faltantes.

El proceso KDD comienza con la carga de las dimensiones a utilizar que en una primera instan-

cia de utilizan todas las variables recolectadas por el Script, es decir se modela de tal manera que el modelo escoja que variables debiese utilizar. Manualmente se realiza el siguiente preproceso:

1. ***Correlación:*** Se eliminan las variables que presentan una correlación superior a 90 % con respecto a otra variable.

2. ***Desviación estándar:*** Se eliminan Variables cuya desviación estándar es inferior al 10 % o en el caso de ser valores nominales el 90 % de las observaciones presentan el mismo valor.

3. ***Valores aberrantes:*** Se realiza una detección y eliminación de valores aberrantes (*outliers*) según la condición de distancia, que no necesariamente corresponde a una distancia euclidiana, esta condición es introducida por Ramaswamy, Rastogi y Shim, que expone un algoritmo que define distancia como la distancia de una observación con respecto a otra según su distancia a sus K vecinos mas cercanos, es decir aplica el modelo KNN (k-th Nearest Neighbor) y se realiza un ranking según este parámetro. En el caso de este trabajo se realiza la medida con respecto a los 10 vecinos mas cercanos.

   Una observación P en el juego de datos es aberrante con respecto a S y D si no mas de k puntos en el juego de datos están a una distancia de D o menos desde P.

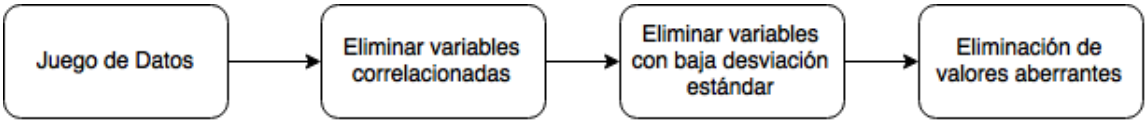


Figura 5.1: Flujo pre procesamiento
Fuente: Elaboración propia

Realizado el preprocesamiento de datos y definido el método de validación se procede a la implementación de algoritmos de minería de datos propiamente tal.

5.2.1. Algoritmos de minería de datos

En la primera aproximación se realiza un modelamiendo solamente considerando la contenido HTML de una página web, lo cual no considera el contenido del texto que se encuentra visible en dicha página ni tampoco información adicional como lo es la seguridad web. Esto tiene dos objetivos, el primero es saber si es posible clasificar una página web considerando solamente contenido, es decir, si existe suficiente evidencia para afirmar que los tipos de páginas web poseen una contenido diferenciable y segundo, si la información adicional que se esta agregando al modelo es relevante al momento de construir un clasificador web.

En el caso de este trabajo la métrica más relevante al momento de escoger un modelo será el *accuracy*, esto se debe a que es la métrica que nos entrega mayor información sobre que tan bien el modelo está clasificando, del total de observaciones cuantas fueron clasificadas correctamente. Notar que debido a que en esta oportunidad se realiza una clasificación entre siete clases la linea base dictada por el azar corresponde a 14,28 %.

Para todos los algoritmos se considera el mismo juego de datos que consiste en 138 observacio-

nes y 20 variables, la diferencia de variables con respecto al capitulo anterior corresponde a que la variable relativa a descripción (nominal) se descompone en 7 variables binarias, lo que suma seis variables extra (Tabla A.1 en anexos).

Con el objetivo de encontrar el mejor resultado posible para cada algoritmo de minería de datos se realiza una optimización de parámetros, es decir se desea encontrar la combinación de parámetros que maximiza la métrica de desempeño, en este caso *accuracy*. Dependiendo del algoritmo de minería de datos, la optimización se puede realizar de forma manual, mediante análisis de sensibilidad o en el caso de algoritmos mas complejos, es necesario automatizar el proceso.
Considerando lo anterior se describen los modelos implementados y sus resultados más importantes.

1. ***Naive Bayes***

   El algoritmo de Naive Bayes es un clasificador supervisado utilizado en el campo de maquinas de aprendizaje que se basa fuertemente en el teorema de Bayes de probabilidades condicionales con una fuerte suposición de independencia entre las variables a considerar[61].
   El modelo Naive Bayes entrega un $accuracy = 52{,}42\,\%$ y es necesario destacar que las clases que presentan mayores problemas al momento de clasificar mediante este modelo son Redes sociales y motores de búsqueda. El algoritmo de Naive Bayes no entrega mucho espacio para

Naive Bayes								
	true entretenimiento	true motor de busqueda	true red social	true noticias	true servicios financieros	true informativa	true ecommerce	class precision
pred. entretenimiento	15	1	2	3	0	2	2	60.00%
pred. motor de busqueda	1	1	2	0	0	1	1	16.67%
pred. red social	2	2	1	1	3	2	0	9.09%
pred. noticias	4	0	0	5	1	1	5	31.25%
pred. servicios financieros	0	0	0	0	3	1	1	60.00%
pred. informativa	3	1	1	1	2	27	3	71.05%
pred. ecommerce	1	0	0	8	1	0	13	56.52%
class recall	57.69%	20.00%	16.67%	27.78%	30.00%	79.41%	52.00%	
Accuracy								**52.42%**

Figura 5.2: Naive bayes
Fuente: Elaboración propia

   calibración puesto que es un modelo probabilístico que se basa en probabilidades condicionales, las cuales no tienen parámetros, sin embargo se puede realizar un ajuste mediante la ”corrección de Laplace”, la cual se encarga de los casos en que se generan probabilidades cero para ciertas variables y estas tienen un alto nivel de influencia en el modelo.
   El modelo corregido entrega un $Accuracy = 56{,}36\,\%$ y existe un comportamiento similar en la *Precision* y *Sensibilidad* (*Recall*) individual de las clases.

Naive Bayes (Corregido)								
	true entretenimiento	true motor de busqueda	true red social	true noticias	true servicios financieros	true informativa	true ecommerce	class precision
pred. entretenimiento	15	1	2	3	0	1	1	65.22%
pred. motor de busqueda	0	1	1	0	0	1	1	25.00%
pred. red social	2	1	1	1	0	1	0	16.67%
pred. noticias	4	0	1	5	2	0	5	29.41%
pred. servicios financieros	0	0	0	0	2	0	1	66.67%
pred. informativa	3	1	1	1	2	26	1	74.29%
pred. ecommerce	1	0	0	8	0	1	12	54.55%
class recall	60.00%	25.00%	16.67%	27.78%	33.33%	86.67%	57.14%	
Accuracy								**56.36%**

Figura 5.3: Naive bayes Corregido
Fuente: Elaboración propia

2. ***Support Vector Machine***

   En la primera aproximación mediante este algoritmo se escoge trabajar bajo el tipo $C-SVM$ y un tipo de kernel de tipo rbf, puesto que las relaciones entre las clases pueden ser no lineales, además es necesario volver a la etapa de transformación en esta instancia pues el algoritmo de SVM no es capaz de manejar variables nominales, por lo que es necesario transformarlas a numéricas, esto es un proceso muy simple, se trata de descomponer una variable nominal en $n-1$ variables binarias de tal forma de asignar solamente un 1 ($true$) a la variable que corresponde a la etiqueta nominal, en el caso de que todas sean cero, es el caso que corresponde a la etiqueta nominal que se excluyo (puesto que son $n-1$). En cuanto a los parámetros del modelo como punto de partida se escoge $C = 0$ y $epsilon = 0{,}001$.
   Tomando en cuenta las consideraciones anteriores se logra un $Accuracy = 37{,}9\,\%$ y se es incapaz de clasificar correctamente ninguna observación de tres clases.

   El algoritmo se calibra empíricamente considerando el punto de partida, para comenzar el proceso se parte con el parámetro C y este se aumenta hasta el punto que su aumento no produce ganancia de $Accuracy$, luego se trabaja con el parámetro epsilon, en un comienzo se intenta disminuir, lo que no produce resultados favorables y luego se aumenta de manera similar a como se trabaja con el parámetro C. Considerar que posiblemente no se ha alcanzado el óptimo puesto que se realizan análisis de Sensibilidad individuales para los parámetros y no conjuntos, sin embargo la ganancia que se obtiene al realizar un ajuste más meticuloso no justifica la dificultad de este, lo que por otro lado no es el objetivo de este trabajo.
   El óptimo encontrado se da fijando $C = 90$ y $epsilon = 0{,}91$.

3. ***Random Forest***

Support Vector Machine (C=0; epsilon= 0.001)								
	true entretenimiento	true motor de busqueda	true red social	true noticias	true servicios financieros	true informativa	true ecommerce	class precision
pred. entretenimiento	9	2	0	5	1	3	4	37.50%
pred. motor de busqueda	1	2	2	0	0	2	0	28.57%
pred. red social	1	1	2	1	0	1	0	33.33%
pred. noticias	4	0	0	5	0	3	6	27.78%
pred. servicios financieros	1	0	1	0	3	2	2	33.33%
pred. informativa	6	0	1	2	5	23	5	54.76%
pred. ecommerce	4	0	0	5	1	0	8	44.44%
class recall	34.62%	40.00%	33.33%	27.78%	30.00%	67.65%	32.00%	
Accuracy								**37.90%**

Figura 5.4: SVM
Fuente: Elaboración propia

Support Vector Machine (C=90; epsilon= 0.91)								
	true entretenimiento	true motor de busqueda	true red social	true noticias	true servicios financieros	true informativa	true ecommerce	class precision
pred. entretenimiento	9	2	0	5	1	3	4	37.50%
pred. motor de busqueda	1	2	2	0	0	2	0	28.57%
pred. red social	1	1	2	1	0	1	0	33.33%
pred. noticias	4	0	0	5	0	3	6	27.78%
pred. servicios financieros	1	0	1	0	3	2	2	33.33%
pred. informativa	6	0	1	2	5	23	5	54.76%
pred. ecommerce	4	0	0	5	1	0	8	44.44%
class recall	34.62%	40.00%	33.33%	27.78%	30.00%	67.65%	32.00%	
Accuracy								**41.94%**

Figura 5.5: SVM Calibrado
Fuente: Elaboración propia

El numero de arboles tiene una incidencia directa en el resultado, generalmente entre más arboles se generen mejor será el resultado y debido a que el modelo se necesita calibrar solo una vez, se generaran tantos arboles como sea computacionalmente factible y exista ganancia seguir agregando arboles.

El criterio para trabajar es ratio de ganancia que tiene una correlación directa con el *Accuracy*, sin embargo es posible modificarse y establecer como criterio que sea efectivamente el *Accuracy* para tener mejores resultados con respecto a dicho criterio.

Lo respectivo a profundidad de los árboles, numero de hojas, las particiones, entre otros son criterios más particulares y se realizará la calibración de forma empírica, por lo que no se detallará el proceso.

Setup Inicial Random Forest	
Numero de arboles	10
Criterio	Ratio de ganancia
Máxima profundidad	20
Confianza	0,25
Ganancia minima	0,1
Minimo de hojas	2
Minimo de hojas para dividir	4
Numero de Prepruning	3
Voting Strategy	Confianza
Pruning	True
Pre-Pruning	True

Figura 5.6: Random Forest, Setup inicial
Fuente: Elaboración propia

El algoritmo entrega un $Accuracy = 45{,}16\,\%$ al entrenar con el setup inicial, considerar que debido a que es un algoritmo que se basa en arboles aleatorios, los resultados son muy variados entre cada iteración, no necesariamente debido a un mejor Setup de parámetros, sin embargo si el numero de arboles aumenta, el factor aleatorio debiese tener menor importancia.

Se optimizan los parámetros de tal forma que el *Accuracy* sea el máximo, el primer parámetro que se fija es la cantidad de arboles a generar en 200, después de dicho número no se obtienen mejores resultados, solo mayor costo computacional, se disminuye la profundidad máxima de los arboles a 18 y se elimina tanto el proceso de pruning como el de pre pruning como se muestra en la figura 5.8.

Tomando en cuenta la optimización de los parámetros del modelo se logra un $Accuracy = 56{,}45\,\%$, sin embargo no fue posible clasificar de forma correcta ninguna observación ni de redes sociales ni motores de búsqueda.

El razonamiento para trabajar con el presente algoritmo es similar al de *RandomForest*, se utiliza el mismo Setup inicial a diferencia del número de árboles puesto que se realiza solo uno, el setup inicial se muestra en la figura 5.10.

El Setup inicial entrega un $Accuracy = 42{,}74\,\%$ al entrenarse con el setup inicial y cabe destacar que es incapaz de clasificar correctamente páginas de redes sociales. Al igual que con el algoritmo de *RandomForest*, los resultados varían entre una iteración y otra y los resultados pueden ser significativamente distintos.

Random Forest Setup inicial								
	true entretenimiento	true motor de busqueda	true red social	true noticias	true servicios financieros	true informativa	true ecommerce	class precision
pred. entretenimiento	12	2	0	3	0	3	2	54.55%
pred. motor de busqueda	0	0	1	0	0	0	0	0.00%
pred. red social	0	0	0	0	0	0	0	0.00%
pred. noticias	2	0	0	1	1	1	1	16.67%
pred. servicios financieros	0	0	0	0	0	0	0	0.00%
pred. informativa	11	3	5	8	8	29	8	40.28%
pred. ecommerce	1	0	0	6	1	1	14	60.87%
class recall	46.15%	0.00%	0.00%	5.56%	0.00%	85.29%	56.00%	
Accuracy								**45.16**

Figura 5.7: Random Forest, Setup inicial
Fuente: Elaboración propia

Setup Optimizado Random Forest	
Numero de arboles	200
Criterio	Accuracy
Máxima profundidad	18
Confianza	-
Ganancia minima	-
Minimo de hojas	-
Minimo de hojas para dividir	-
Numero de Prepruning	-
Voting Strategy	Confianza
Pruning	False
Pre-Pruning	False

Figura 5.8: Random Forest, Setup Optimizado
Fuente: Elaboración propia

La optimización de los parámetros se realiza de forma empírica, también de forma muy similar a como se realiza con el algoritmo de *RandomForest*, sin embargo el óptimo se alcanza con un setup distinto como se muestra en la figura 5.12.

El método $DecisionTreeClassificator$ entrega un $Accuracy = 52{,}42\,\%$ una vez que se ha optimizado el set de parámetros y si bien no entrega buenos resultados en redes sociales, no son clasificadas erróneamente en su totalidad.

4. ***Artificial Neural Network***

Una red neuronal es un algoritmo tan complejo como el desarrollador quiere que sea,en este caso se realiza una red neuronal artificial de tipo MLP que se entrena bajo *backpropagation* y para la optimización se realizará en torno a tres parámetros:

Random Forest Setup Optimizado								
	true entretenimiento	true motor de busqueda	true red social	true noticias	true servicios financieros	true informativa	true ecommerce	class precision
pred. entretenimiento	19	2	5	5	0	2	1	55.88%
pred. motor de busqueda	0	0	0	0	0	0	0	0.00%
pred. red social	0	0	0	0	0	0	0	0.00%
pred. noticias	0	0	0	1	0	0	2	33.33%
pred. servicios financieros	0	0	0	0	2	0	1	66.67%
pred. informativa	5	3	1	4	7	31	4	56.36%
pred. ecommerce	2	0	0	8	1	1	17	58.62%
class recall	73.08%	0.00%	0.00%	5.56%	20.00%	91.18%	68.00%	
Accuracy								**56.45%**

Figura 5.9: Random Forest, Setup Optimizado
Fuente: Elaboración propia

Setup Inicial Decision Tree	
Criterio	Ratio de ganancia
Máxima profundidad	20
Confianza	0,25
Ganancia minima	0,1
Minimo de hojas	2
Minimo de hojas para dividir	4
Numero de Prepruning	3
Voting Strategy	Confianza
Pruning	True
Pre-Pruning	True

Figura 5.10: Decision Tree Classificator, Setup inicial
Fuente: Elaboración propia

(a) ***Ciclos de entrenamiento:*** Corresponde a cuantos ciclos se utilizaran para optimizar los pesos de cada neurona en la red mediante ”back propagation”.

(b) ***Tasa de aprendizaje:*** Determina cuando cambian los pesos de las neuronas en cada ciclo de entrenamiento.

(c) ***Momentum:*** Es una tasa que agrega una fracción de la actualización de pesos del ciclo de entrenamiento anterior a la actual con el objetivo de prevenir optimos locales en cierta medida.

El setup inicial para la primera iteración con este algoritmo se muestra en la tabla 5.14.

Decision Tree Setup inicial								
	true entretenimiento	true motor de busqueda	true red social	true noticias	true servicios financieros	true informativa	true ecommerce	class precision
pred. entretenimiento	12	2	0	10	2	4	4	35.29%
pred. motor de busqueda	3	1	1	0	0	0	1	16.67%
pred. red social	0	0	0	1	4	4	0	0.00%
pred. noticias	4	0	1	2	0	1	1	22.22%
pred. servicios financieros	0	0	2	0	1	5	1	11.11%
pred. informativa	6	2	2	0	3	20	1	58.82%
pred. ecommerce	1	0	0	5	0	0	17	73.91%
class recall	46.15%	20.00%	0.00%	11.11%	10.00%	58.82%	68.00%	
Accuracy								**42.74%**

Figura 5.11: Decision Tree Classificator
Fuente: Elaboración propia

Setup Optimizado Decision Tree	
Criterio	Accuracy
Máxima profundidad	10
Confianza	-
Ganancia minima	-
Minimo de hojas	-
Minimo de hojas para dividir	-
Numero de Prepruning	-
Pruning	False
Pre-Pruning	False

Figura 5.12: Decision Tree Classificator, Setup Optimizado
Fuente: Elaboración propia

El algoritmo de $RedNeuronalArtificial$ entrega un $Accuracy = 52{,}67\,\%$ considerando el setup inicial, el cual no es capaz de clasificar correctamente motores de búsqueda, sin embargo dicha clase es la que posee el menor número de observaciones.

El algoritmo se calibra manualmente considerando los tres parámetros anteriormente expuestos, generalmente el número de ciclos lleva a mejores resultados puesto que hay un mayor número de oportunidades de aprendizaje, para los otros dos parámetros la combinación que entrega el óptimo depende de cada problema y no existe forma explícita para estimarlos[62]. Con el objetivo de estimar de la mejor manera posible los parámetros para la red neuronal se intenta optimizar los parámetros $Tasadeaprendizaje$ y $Momentum$ para un bajo número de $ciclosdeentrenamiento$ puesto que su tiempo de compilación disminuye considerablemente a medida que disminuyen los ciclos de entrenamiento, las pruebas se muestran en la tabla 5.15.

Decision Tree Setup Optimizado								
	true entretenimiento	true motor de busqueda	true red social	true noticias	true servicios financieros	true informativa	true ecommerce	class precision
pred. entretenimiento	14	0	4	3	0	2	2	56.00%
pred. motor de busqueda	0	2	0	0	0	0	0	100.00%
pred. red social	3	1	1	1	0	1	0	14.29%
pred. noticias	3	0	1	6	0	3	4	35.29%
pred. servicios financieros	0	0	0	1	4	1	1	57.14%
pred. informativa	3	2	0	2	5	25	5	59.52%
pred. ecommerce	3	0	0	5	1	2	13	54.17%
class recall	53.85%	40.00%	16.67%	33.33%	40.00%	73.53%	52.00%	
Accuracy								**52.42%**

Figura 5.13: Decision Tree Classificator Optimizado
Fuente: Elaboración propia

ANN setup inicial	
Ciclos de entrenamiento	500
Tasa de aprendizaje	0,3
Momentum	0,2

Figura 5.14: ANN Setup inicial
Fuente: Elaboración propia

Sin embargo el óptimo encontrado para 200 $ciclos de entrenamiento$ no puede ser extrapolado para un número distinto de ciclos, por ejemplo la combinación $[0{,}04; 0{,}05]$ que entrega el óptimo para 200 ciclos con un desempeño de $Accuracy = 57{,}25\,\%$, entrega un valor subóptimo de $Accuracy = 53{,}44\,\%$ para 750 ciclos, valor que debiese incrementar al aumentar el número de ciclos de entrenamiento o al menos mantenerse, es por esta razón que se concluye que los óptimos no son transferibles para distintos números de ciclos. Se intenta una nueva optimización para 750 ciclos pero es imposible replicar el procedimiento anterior debido a los tiempos de compilación elevados para dicha cantidad de ciclos, sin embargo se utiliza una librería que logra automatizar dicho proceso de optimización mediante una variación de parámetros de forma lineal y tras cada iteración se comparan los resultados de *Accuracy*, este proceso es muy demandante tanto de tiempo como de poder computacional, sin embargo se logra un óptimo de $Accuracy = 58{,}78\,\%$ considerando los parámetros de la tabla 5.17.

El modelo que clasifica de mejor manera páginas web es el de red neuronal artificial para la primera aproximación de este problema, este algoritmo fue capaz de clasificar páginas web con $Accuracy = 58{,}78\,\%$. Es necesario mencionar que para este problema no es relevante la carga computacional que requiere cada modelo, puesto que dicho proceso solo se realiza una vez al momento de calibrar los pesos de las variables.

Red Neuronal Artificial Setup Inicial								
	true entretenimiento	true motor de busqueda	true red social	true noticias	true servicios financieros	true informativa	true ecommerce	class precision
pred. entretenimiento	16	0	3	2	0	3	0	66.67%
pred. motor de busqueda	0	0	0	0	0	1	0	0.00%
pred. red social	3	0	3	1	0	1	0	37.50%
pred. noticias	3	1	0	4	3	3	3	23.53%
pred. servicios financieros	0	0	0	1	5	4	1	45.45%
pred. informativa	4	2	0	7	5	27	3	56.25%
pred. ecommerce	2	0	0	5	0	1	14	63.64%
class recall	57.14%	0.00%	50.00%	20.00%	38.46%	67.50%	66.67%	
Accuracy								**52.67%**

Figura 5.15: ANN Setup inicial
Fuente: Elaboración propia

ANN optimización 200 ciclos de entrenamiento						
Ciclos de entrenamiento	200	200	200	200	200	200
Tasa de aprendizaje	0,3	0,2	0,15	0,02	0,05	0,04
Momentum	0,2	0,2	0,1	0,02	0,05	0,05
Accutacy	50.4%	52.67%	53.44%	53.44%	56.49%	57.25%

Figura 5.16: ANN Optimización
Fuente: Elaboración propia

ANN Setup optimizado	
Ciclos de entrenamiento	700
Tasa de aprendizaje	0,5
Momentum	0

Figura 5.17: ANN Setup Optimizado
Fuente: Elaboración propia

A continuación se muestra un resumen de todos los modelos mencionados en este capítulo junto con su respectivo desempeño en su etapa inicial y optimizada (ver figura 6.3).

Considerando el trabajo y los resultados anteriores, se propone la incorporación de variables relacionadas al contenido de la página web, que si bien no es parte de clasificación por contenido HTML, se cree que aportará gran valor al momento de clasificar páginas web de forma correcta y por consiguiente ayudará a los objetivos fundamentales de este trabajo.

Red Neuronal Artificial Setup Optimizado								
	true entretenimiento	true motor de busqueda	true red social	true noticias	true servicios financieros	true informativa	true ecommerce	class precision
pred. entretenimiento	17	0	1	2	0	4	1	68.00%
pred. motor de busqueda	1	0	0	0	0	0	0	0.00%
pred. red social	2	0	3	1	0	0	0	50.00%
pred. noticias	3	1	1	9	3	3	3	39.13%
pred. servicios financieros	0	0	0	2	7	5	1	46.67%
pred. informativa	4	1	1	4	3	28	3	63.64%
pred. ecommerce	1	1	0	2	0	0	13	76.47%
class recall	60.71%	0.00%	50.00%	45.00%	53.85%	70.00%	61.90%	
Accuracy								**58.78%**

Figura 5.18: ANN Setup Optimizado
Fuente: Elaboración propia

Resumen modelamiento según contenido HTML					
Algoritmo	Naive Bayes	SVM	Random Forest	Decision Tree	ANN
Accuracy Inicial	52,42%	37,90%	45,16%	42,74%	52,67%
Accuracy Optimizado	**56,36%**	**41,94%**	**56,45%**	**52,42%**	**58,78%**

Figura 5.19: Resumen modelos
Fuente: Elaboración propia

5.3. Minería de textos

La minería de textos o Text Data Mining tiene por objetivo ayudar a los usuarios a encontrar documentos o información en ellos que satisfagan sus necesidades[38].

La razón de la implementación de TDM en este trabajo de título reside en el creciente interés por el tema en la investigación actual y los buenos resultados que se han logrado al analizar grandes fuentes de datos tanto de texto estructurado como de no estructurado[63], las cuales han sido capaces de explicar patrones que hasta el momento no había sido posible considerando el escenario de los experimentos.

La implementación del proceso se realiza utilizando el software RapidMiner Studio 7.1 y la extensión WEKA de este. La extensión WEKA provee todas las herramientas necesarias para realizar procesamiento de texto de muy buena forma, pasando por todas las etapas que se detallan anteriormente.
Existen tres consideraciones que son importantes de mencionar que son particulares de este problema y pueden hacer una diferencia entre el proceso estandarizado que se propone y la implementación de este.

1. Terminada la etapa de eliminación de *Stopwords* se procede a realizar un segundo proceso de filtrado, se eliminan todas las palabras cuyo largo es menor a 4 caracteres, puesto que en su gran mayoría agregan muy bajo valor, complican innecesariamente el modelo y en algunos casos generan ruido y afectan el *Accuracy* del algoritmo. Se realiza un proceso similar para palabras cuyo largo supera los 25 caracteres por razones similares, dichas palabras generalmente son múltiples palabras que no fueron separadas por espacio, las cuales no tienen valor para este estudio.

2. En la etapa de generación del *WordPage* Vector se realiza el proceso de TF-IDF, sin embargo se realiza otro proceso de filtrado, primero se ordenan todas las palabras del documento por frecuencia (Frecuency ranking) y se eliminan todas las palabras cuya frecuencia es mayor al 95 % del total del ranking (total de palabras) o menor al 5 % del mismo[1].

3. El proceso de *Steming* es particularmente difícil para este problema debido a existen múltiples idiomas a tratar y dentro de un mismo idioma, muchas variaciones de palabras dependiendo del país al que pertenece el sitio web (y por consiguiente la página web a clasificar). En un principio este problema se intenta abordar realizando una traducción del texto.

   (a) La primera solución a este problema fue intentar utilizar la API que provee Google para traducir texto, sin embargo para textos de la extensión que se intenta manejar no es un servicio gratis, por lo que para los objetivos de este trabajo no podrá ser utilizada.

   (b) La segunda solución tiene mucha relación con la primera, se realiza un minado de la web sobre la aplicación que provee Google para traducciones online dentro de su buscador web, sin embargo por construcción de su buscador entrega dos problemas importantes, primero existen un límite de palabras que se sobrepasa en cada iteración (cerca de 3000 caracteres) por lo que fue necesario cortar el texto, traducir y luego pegarlo antes de comenzar el proceso de minería de textos, sin embargo esto generó nuevas palabras que fueron resultados del corte y pérdida de palabras que podrían entregar valor. Este proceso podría refinarse de tal manera de cortar en el ultimo espacio antes de alcanzar los 3000 caracteres pero no se justifica debido al punto posterior.

      El segundo problema relativo al traductor mediante web crawling es que un número considerable de páginas no son posibles de traducir debido a que en el texto visible existen caracteres que no son letras ni símbolos de puntuación, lo que produce un error al traducir, por lo que la única solución viable para esto fue la implementación de un bloque try-catch que opera intentando traducir y en el caso que no es posible, dejar el texto como fue entregado y seguir a la siguiente página web.

      Considerando las complicaciones descritas se descarta la traducción de texto para el presente trabajo de titulo y se deja propuesto como trabajo futuro.

[1] Este proceso es conocido como pruning

La segunda aproximación fue realizar el proceso de Steming para el idioma más recurrente en el juego de datos, es decir para español, esto se realiza con la ayuda de la librería snowball que facilita algoritmos de Steming para distintos idiomas y es de las pocas que incluye al español. Dependiendo de los resultados se evaluará de la implementación del proceso de Steming, puesto que podría restar valor al momento de modelar los datos y realizar la clasificación web.

5.3.1. Algoritmos de minería de datos en text mining

La implementación de los algoritmos de minería de datos se realiza de manera similar a como se hace en la sección anterior. El proceso comienza con la carga del vector de características, con el agregado de la variable de texto visible en la página web, se realiza el mismo pre-procesamiento sobre el set de variables inicial (eliminación de observaciones por correlación, desviación estándar y valores aberrantes), luego se trabaja solo con la variable relativa a texto visible y se realiza el proceso de TF-IDF descrito anteriormente y finaliza con la implementación de algoritmos de minería de datos, los cuales son evaluados por validación cruzada (ver figura 5.20).

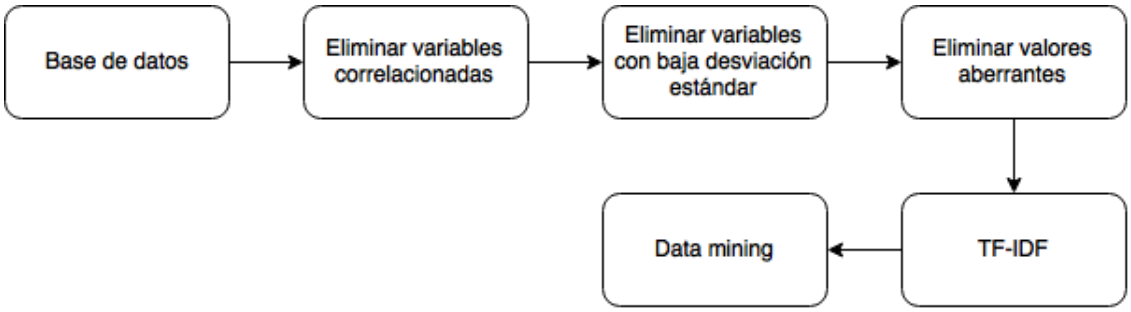


Figura 5.20: Proceso de text mining
Fuente: Elaboración propia

Para la implementación de los algoritmos de minería de datos se utilizan las mismas 138 observaciones, sin embargo debido a la adición de esta nueva variable el set de variables aumenta de 20 variables a 39617 (En la figura A.2 del anexo se muestra un subconjunto del vector de palabras). A continuación se muestran los algoritmos implementados y sus principales resultados.

1. ***Naive Bayes***

   La linea base de este algoritmo corresponde a TF-IDF que considera un pruning que elimina valores bajo el 5 % de frecuencia y sobre un 95 % de frecuencia con respecto al total del ranking y un *NaiveBayes* sin corrección de *Laplace*. El caso base del algoritmo *NaiveBayes* entrega un Accuracy de 73.60 % (Figura 5.21).

   A continuación se quiere hacer la prueba de si mejoran los resultados al eliminar el proceso de Steming, para lo cual se implementa el mismo modelo salvo dicho proceso. El modelo

Naive Bayes								
	true entretenimiento	true motor de busqueda	true red social	true noticias	true servicios financieros	true informativa	true ecommerce	class precision
pred. entretenimiento	26	0	1	0	0	1	2	86.67%
pred. motor de busqueda	0	0	0	0	0	0	0	0.00%
pred. red social	0	1	3	1	0	0	0	60.00%
pred. noticias	1	1	0	13	2	6	1	54.17%
pred. servicios financieros	1	0	0	0	8	1	1	72.73%
pred. informativa	0	1	2	6	0	29	2	72.50%
pred. ecommerce	0	0	0	0	0	2	13	86.67%
class recall	92.86%	0.00%	50.00%	65.00%	80.00%	74.36%	68.42%	
Accuracy								**73.60%**

Figura 5.21: Naive Bayes con Steming
Fuente: Elaboración propia

arroja un $Accuracy = 75{,}20\,\%$ (Figura 5.22) al eliminar el proceso de *Steming*, lo cual representa una mejora de $1{,}8\,\%$ con respecto al original, si bien no es concluyente, para este algoritmo se realizará la optimización sin considerar *Steming*.

Naive Bayes								
	true entretenimiento	true motor de busqueda	true red social	true noticias	true servicios financieros	true informativa	true ecommerce	class precision
pred. entretenimiento	22	0	2	0	0	1	2	81.48%
pred. motor de busqueda	0	1	0	0	0	0	0	100.00%
pred. red social	2	1	4	1	0	0	0	50.00%
pred. noticias	1	0	0	16	2	6	1	61.54%
pred. servicios financieros	1	0	0	0	8	1	1	72.73%
pred. informativa	1	1	0	3	0	30	2	81.08%
pred. ecommerce	1	0	0	0	0	1	13	86.67%
class recall	78.57%	33.33%	66.67%	80.00%	80.00%	76.92%	68.42%	
Accuracy								**75.20%**

Figura 5.22: Naive Bayes sin Steming
Fuente: Elaboración propia

Para finalizar se optimiza el modelo considerando el proceso de *pruning* y la corrección de *Laplac*e, sin embargo las variaciones en el proceso de pruning solo entregaron resultados menos favorables, por lo que se considera optimizado en los parámetros iniciales, la corrección de *Laplac*e por otro lado mejora bastante la situación base llegando a un $Accuracy = 77{,}54\,\%$ (Figura 5.23), lo que corresponde a una mejora de 2,43 % con respecto a la situación del caso no corregido.

Naive Bayes								
	true entretenimiento	true motor de busqueda	true red social	true noticias	true servicios financieros	true informativa	true ecommerce	class precision
pred. entretenimiento	22	0	2	0	0	2	1	81.48%
pred. motor de busqueda	4	1	1	1	2	1	1	9.09%
pred. red social	1	1	4	0	0	1	0	57.14%
pred. noticias	1	0	0	19	1	6	0	70.37%
pred. servicios financieros	0	1	0	0	9	1	0	81.82%
pred. informativa	1	0	0	0	1	33	1	91.67%
pred. ecommerce	0	0	0	0	0	0	19	100.00%
class recall	75.86%	33.33%	57.14%	95.00%	69.23%	75.00%	86.36%	
Accuracy								**77.54%**

Figura 5.23: Naive Bayes sin Steming y corrección de Laplace
Fuente: Elaboración propia

2. ***Support Vector Machine***

La linea base de este algoritmo corresponde a $TF - IDF$ que considera un pruning que elimina valores bajo el 5 % de frecuencia y sobre un 95 % de frecuencia con respecto al total del ranking y un *SVM* de clasificación múltiple que utiliza un kernel de tipo *RBF* con $gamma = 0$ y $C = 0$. Este caso base entrega un $Accuracy = 33{,}33\,\%$ (figura 5.24).

El paso siguiente es eliminar el proceso de Steming, lo que entrega resultados iguales al modelo con *Steming* (figura 5.25), sin embargo es necesario destacar que ambos son resultados bastante poco razonables considerando los obtenidos por *NaiveBayes* utilizando el mismo juego de datos y variables, además debido a que los resultados no mejoran, ya se tiene indicios fuertes de que el proceso de Steming no estaría aportando valor al modelo, más aún posiblemente lo perjudica, con el fin de comprobar la veracidad de esta afirmación se hará una última prueba en el modelo que tuvo los mejores resultados con el set de variables anterior: *ArtificialNeuralNetwork*.

Tomando en cuenta que considerar el proceso de Steming no estaría mejorando el desempeño del algoritmo, no se considera para la optimización. La optimización de parámetros para

Support Vector Machine								
	true entretenimiento	true motor de busqueda	true red social	true noticias	true servicios financieros	true informativa	true ecommerce	class precision
pred. entretenimiento	4	0	0	13	0	3	9	13.79%
pred. motor de busqueda	0	0	0	0	0	0	0	0.00%
pred. red social	0	0	0	0	0	0	0	0.00%
pred. noticias	7	0	0	3	1	1	1	23.08%
pred. servicios financieros	0	0	0	0	0	0	0	0.00%
pred. informativa	14	3	7	4	12	39	12	42.86%
pred. ecommerce	4	0	0	0	0	1	0	0.00%
class recall	13.79%	0.00%	0.00%	15.00%	0.00%	88.64%	0.00%	
Accuracy								**33.33%**

Figura 5.24: Support Vector Machine con Steming
Fuente: Elaboración propia

Support Vector Machine								
	true entretenimiento	true motor de busqueda	true red social	true noticias	true servicios financieros	true informativa	true ecommerce	class precision
pred. entretenimiento	4	0	0	13	0	3	9	13.79%
pred. motor de busqueda	0	0	0	0	0	0	0	0.00%
pred. red social	0	0	0	0	0	0	0	0.00%
pred. noticias	7	0	0	3	1	1	1	23.08%
pred. servicios financieros	0	0	0	0	0	0	0	0.00%
pred. informativa	14	3	7	4	12	39	12	42.86%
pred. ecommerce	4	0	0	0	0	1	0	0.00%
class recall	13.79%	0.00%	0.00%	15.00%	0.00%	88.64%	0.00%	
Accuracy								**33.33%**

Figura 5.25: Support Vector Machine sin Steming
Fuente: Elaboración propia

este modelo se realiza de forma similar a como se optimizaron los parámetros para la red neuronal artificial para el set de variables sin considerar contenido de texto, se realiza una optimización por grilla, que considera los valores iniciales como punto de partida y realiza análisis de Sensibilidad en los parámetros, realizando variaciones lineales sobre estos a medida que aumenta el desempeño del modelo. El conjunto de parámetros optimizados considera

gamma $= 7$ y $C = 43123$ y *pruning* en el proceso $TF - IDF$ que considera eliminar valores bajo el 5 % de frecuencia y sobre un 95 % de frecuencia con respecto al total del ranking lo que resulta en un *Accuracy* $= 35{,}51\,\%$ (figura 5.26).

Support Vector Machine								
	true entretenimiento	true motor de busqueda	true red social	true noticias	true servicios financieros	true informativa	true ecommerce	class precision
pred. entretenimiento	0	0	0	0	0	1	0	0.00%
pred. motor de busqueda	0	0	0	0	0	0	0	0.00%
pred. red social	0	0	0	1	0	0	0	0.00%
pred. noticias	0	0	1	2	0	0	0	66.67%
pred. servicios financieros	0	0	0	0	2	0	1	66.67%
pred. informativa	29	3	6	17	10	43	19	33.86%
pred. ecommerce	0	0	0	0	1	0	2	66.67%
class recall	0.00%	0.00%	0.00%	10.00%	15.38%	97.73%	9.09%	
Accuracy								**35.51%**

Figura 5.26: Support Vector Machine Optimizado
Fuente: Elaboración propia

3. ***Random Forest***

En el presente algoritmo se consideran los aprendizajes obtenido en el modelamiento sin considerar contenido de textos, por lo que el criterio para la linea base considera *Accuracy* como principal criterio de desempeño y mantener los valores de linea base tanto de pruning como de pre-pruning de la linea base propuesta para el modelamiento anterior (ver figura 5.27).

Setup inicial Decision Tree	
Número de árboles	10
Criterio	Accuracy
Máxima profundidad	20
Confianza	0,25
Ganancia mínima	0,1
Mínimo de hojas	2
Mínimo de hojas para dividir	4
Número de prepruning	3
Voting strategy	Confianza
Pruning	True
Pre-pruning	Tue

Figura 5.27: Random Forest Setup inicial
Fuente: Elaboración propia

El setup que se describe anteriormente, en conjunto al proceso de TF-IDF que considera Steming, además de un proceso de pruning que considera eliminar valores bajo el 5 % de

frecuencia y sobre un 95 % de frecuencia con respecto al total del ranking consiste en la nueva linea base de este algoritmo que resulta en un $Accuracy = 31{,}88\,\%$ (ver figura 5.29)

Random Forest								
	true entretenimiento	true motor de busqueda	true red social	true noticias	true servicios financieros	true informativa	true ecommerce	class precision
pred. entretenimiento	0	0	0	0	0	0	0	0.00%
pred. motor de busqueda	0	0	0	0	0	0	0	0.00%
pred. red social	0	0	0	0	0	0	0	0.00%
pred. noticias	0	0	0	0	0	0	0	0.00%
pred. servicios financieros	0	0	0	0	0	0	0	0.00%
pred. informativa	29	3	7	20	13	44	22	31.88%
pred. ecommerce	0	0	0	0	0	0	0	0.00%
class recall	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%	0.00%	
Accuracy								**31.88%**

Figura 5.28: Random Forest
Fuente: Elaboración propia

Estos resultados se contrastan con los resultados obtenidos por este mismo algoritmos bajo el mismo setup con la salvedad de eliminar el proceso de Steming con el objetivo de validar el valor de este para estudios posteriores considerando las condiciones actuales. El algoritmo que no considera Steming alcanza un $Accuracy = 31{,}88\,\%$ (ver figura 5.29), que es el mismo resultado que el modelo que si considera Steming, es más, las categorías se clasifican de la misma forma y solo es capaz de clasificar páginas informativas, es por esto que se decide no utilizar este proceso en estudios posteriores y se procede a la optimización de este algoritmo.

El algoritmo se optimiza aumentando el número de árboles desde 10 a 100 y considerando todos los parámetros que involucran a los procesos de *pruning* y $pre-pruning$, sin embargo fue imposible realizar el procesamiento de este algoritmo y menos aún ningún tipo de optimización, considerando las condiciones actuales de poder computacional y los requerimientos necesarios para lograrlo.

Se propone realizar una segunda parte en el preprocesamiento con el objetivo de ser capaz de realizar este algoritmo e intentar replicarlo para el siguiente algoritmo que aumenta los requerimientos computacionales, a continuación se describe el proceso de preprocesamiento utilizado.

(a) ***Correlación:*** Se eliminan las variables que presentan una correlación superior a 90 % con respecto a otra variable.

(b) ***Desviación estándar:*** Se eliminan Variables cuya desviación estándar es inferior al

Random Forest								
	true entretenimiento	true motor de busqueda	true red social	true noticias	true servicios financieros	true informativa	true ecommerce	class precision
pred. entretenimiento	0	0	0	0	0	0	0	0.00%
pred. motor de busqueda	0	0	0	0	0	0	0	0.00%
pred. red social	0	0	0	0	0	0	0	0.00%
pred. noticias	0	0	0	0	0	0	0	0.00%
pred. servicios financieros	0	0	0	0	0	0	0	0.00%
pred. informativa	29	3	7	20	13	44	22	31.88%
pred. ecommerce	0	0	0	0	0	0	0	0.00%
class recall	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%	0.00%	
Accuracy								**31.88%**

Figura 5.29: Random Forest
Fuente: Elaboración propia

10 % o en el caso de ser valores nominales el 90 % de las observaciones presentan el mismo valor.

(c) ***Valores aberrantes:*** Se realiza una detección y eliminación de valores aberrantes (outliers) según la condición de distancia.

(d) ***TF-IDF:*** Creación de variables relativas a los n-gramas provenientes del proceso TF-IDF.

(e) ***Correlación:*** Se eliminan las variables que presentan una correlación superior a 90 % con respecto a otra variable, considerando las nuevas variables producto de TF-IDF.

(f) ***Desviación estándar:*** Se eliminan Variables cuya desviación estándar es inferior al 10 % o en el caso de ser valores nominales el 90 % de las observaciones presentan el mismo valor, considerando las nuevas variables producto de TF-IDF.

(g) ***Valores aberrantes:*** Se realiza una detección y eliminación de valores aberrantes (outliers) según la condición de distancia, considerando las nuevas variables producto de TF-IDF.

(h) ***Transformación de atributos nominales a numéricos:*** El siguiente proceso no es capaz de manejar atributos nominales por lo que una transformación es necesaria.

(i) ***Principal Component Analysis (PCA):*** proceso para reducir dimensionalidad a través de matriz de co-varianza, esto es que se reduzca la cantidad de atributos redundantes de modo que la covarianza entre atributos no supere el 96 %.

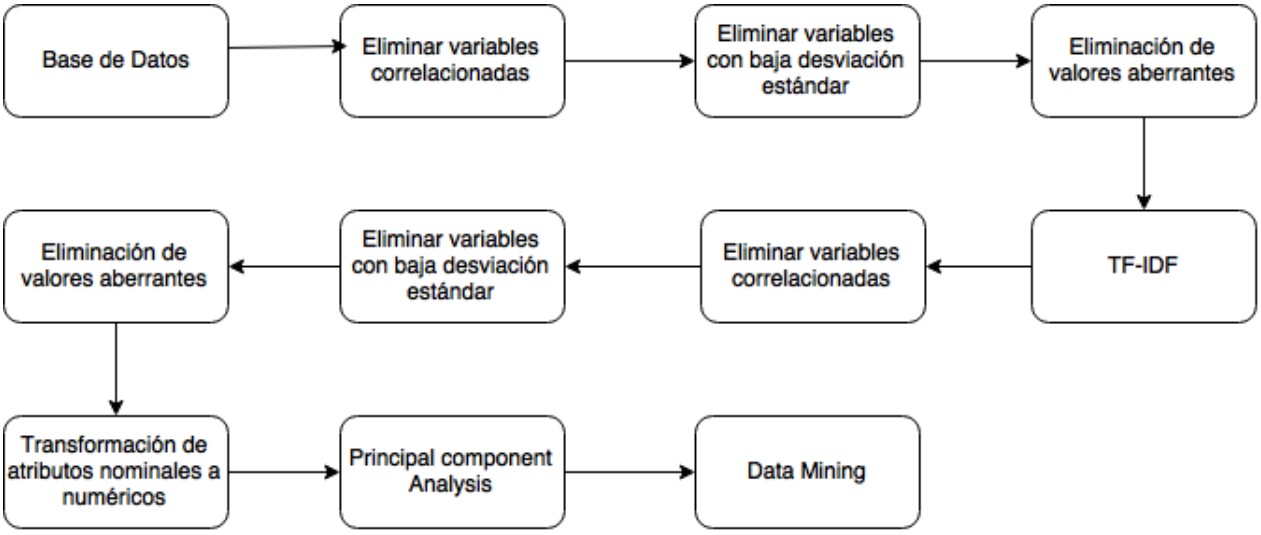


Figura 5.30: Preprocesamiento Random Forest
Fuente: Elaboración propia

El preprocesamiento recién descrito es capaz de reducir la dimensionalidad del problema de 39617 atributos a 2092 (ver anexos B.1) y de esta forma, es posible desarrollar y aplicar el algoritmo de forma correcta, en cuya optimización final se llega a un número de árboles de decisión de 100, lo que concluye en un $Accuracy = 36{,}96\,\%$ (ver figura 5.31).

4. ***Decision Tree***

En el algoritmo de Decision Tree se realizan ciertas consideraciones, primero para la linea base se considera el *Accuracy* como criterio de desempeño en vez de radio de ganancia (que fue la linea base en el modelamiento anterior), puesto que es el criterio que se utilizará finalmente para medir el desempeño de este trabajo, además se comprobó en el modelamiento anterior (sin utilizar contenido textual web) que utilizando dicho criterio se obtienen mejores resultados, con respecto al *pruning* es considerado en el modelo base, debido a que no se sabe con certeza si este tiene un impacto negativo o positivo en el desempeño del modelo, los demás parámetros se consideran igual que la linea base del modelamiento anterior utilizando este modelo (ver figura 5.32). El caso base de este modelo tampoco considera Steming, puesto que en base a los estudios anteriores se considera que en el mejor de los casos no agrega valor al algoritmo.

El algoritmo Decision Tree entrega un $Accuracy = 55{,}8\,\%$ (ver figura 5.33) en el caso base tomando en cuenta las consideraciones expuestas.

Random Forest								
	true entretenimiento	true motor de busqueda	true red social	true noticias	true servicios financieros	true informativa	true ecommerce	class precision
pred. entretenimiento	2	0	1	7	1	1	7	10.53%
pred. motor de busqueda	0	1	1	0	0	1	0	33.33%
pred. red social	1	1	2	1	0	0	0	40.00%
pred. noticias	9	0	1	8	0	1	1	40.00%
pred. servicios financieros	0	0	0	0	0	4	2	0.00%
pred. informativa	10	1	2	3	11	34	8	49.28%
pred. ecommerce	7	0	0	1	1	3	4	25.00%
class recall	6.90%	33.33%	28.57%	40.00%	0.00%	77.27%	18.18%	
Accuracy								**36.96%**

Figura 5.31: Random Forest Optimizado
Fuente: Elaboración propia

Setup inicial Decision Tree	
Criterio	Accuracy
Máxima profundidad	20
Confianza	0,25
Ganancia mínima	0,1
Mínimo de hojas	2
Mínimo de hojas para dividir	4
Número de prepruning	3
Voting strategy	Confianza
Pruning	True
Pre-pruning	Tue

Figura 5.32: Setup inicial Decision Tree
Fuente: Elaboración propia

La optimización se realiza con respecto a la profundidad máxima del árbol de decisión y con respecto a los procesos de *pruning*, *prepruning* y todos los parámetros que involucran estos últimos dos (ver figura 5.34), notar que el algoritmo optimizado no considera el proceso de *pruning*. La optimización también implica una optimización del pruning con respecto al proceso de $TF - IDF$, el cual mantiene su óptimo en eliminar valores bajo el 5 % de frecuencia y sobre un 95 % de frecuencia con respecto al total del ranking.

La optimización de parámetros en el algoritmo produce una mejora de 1.45 % con respecto al modelo no optimizado llegando a un $Accuracy = 57{,}25\,\%$ (ver figura 5.35)

5. ***Red Neuronal Artificial***

En el presente algoritmo se utilizará como parámetros iniciales 500 ciclos de entrenamiento, tasa de aprendizaje = 0,3 y $momentum = 0,2$ (ver figura 5.36) y al igual que en los casos

Decisión Tree Classificator								
	true entretenimiento	true motor de busqueda	true red social	true noticias	true servicios financieros	true informativa	true ecommerce	class precision
pred. entretenimiento	12	0	0	4	3	5	4	42.86%
pred. motor de busqueda	0	0	0	0	0	0	0	0.00%
pred. red social	1	1	0	1	0	0	0	0.00%
pred. noticias	4	0	0	10	0	1	1	62.50%
pred. servicios financieros	1	0	0	0	6	2	2	54.55%
pred. informativa	9	2	7	2	2	36	2	60.00%
pred. ecommerce	2	0	0	3	2	0	13	65.00%
class recall	41.38%	0.00%	0.00%	50.00%	46.15%	81.82%	59.09%	
Accuracy								**55.80%**

Figura 5.33: Decision Tree caso base
Fuente: Elaboración propia

Setup inicial Decision Tree	
Criterio	Accuracy
Máxima profundidad	21
Confianza	-
Ganancia mínima	0,05
Mínimo de hojas	2
Mínimo de hojas para dividir	4
Número de prepruning	3
Voting strategy	Confianza
Pruning	False
Pre-pruning	Tue

Figura 5.34: Decision Tree caso base
Fuente: Elaboración propia

anteriores se considerará como métrica primordial el *Accuracy*.

Tomando en cuenta la complejidad del juego de datos y la carga computacional que implica el algoritmo de Red Neuronal Artificial, no es posible implementar dicho algoritmo, debido a restricciones de poder computacional. Se propone como solución reducir la dimensionalidad del problema a través de un preprocesamiento más exhaustivo, considerando el mismo proceso descrito en el algoritmo de *RandomForest*.

Considerando el nuevo preprocesamiento se calcula el Accuracy bajo los parámetros de la linea base la cual resulta de 38.41 % (ver figura 5.38).

Decisión Tree Classificator								
	true entretenimiento	true motor de busqueda	true red social	true noticias	true servicios financieros	true informativa	true ecommerce	class precision
pred. entretenimiento	12	0	0	4	3	5	4	42.86%
pred. motor de busqueda	0	0	0	0	0	0	0	0.00%
pred. red social	1	1	2	1	0	0	0	40.00%
pred. noticias	4	0	0	10	0	1	1	62.50%
pred. servicios financieros	1	0	1	0	6	2	2	50.00%
pred. informativa	9	2	4	2	2	36	2	63.16%
pred. ecommerce	2	0	0	3	2	0	13	65.00%
class recall	41.38%	0.00%	28.57%	50.00%	46.15%	81.82%	59.09%	
Accuracy								**57.25%**

Figura 5.35: Decision Tree caso base
Fuente: Elaboración propia

ANN setup inicial	
Ciclos de entrenamiento	500
Tasa de aprendizaje	0,3
Momentum	0,2

Figura 5.36: Decision Tree caso base
Fuente: Elaboración propia

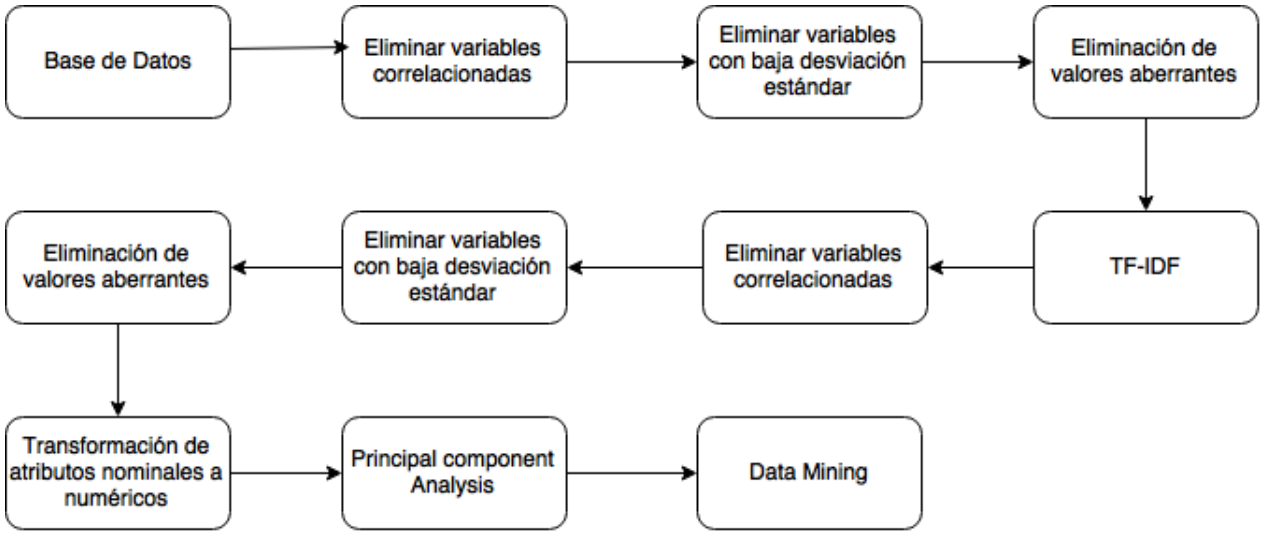


Figura 5.37: Preprocesamiento Red Neuronal Artificial
Fuente: Elaboración propia

A continuación se procede a realizar una optimización del algoritmo con respecto a los parámetros *LearningRate* y *Momentum*, esta optimización no se realiza de forma manual, si

Red Neuronal Artificial								
	true entretenimiento	true motor de busqueda	true red social	true noticias	true servicios financieros	true informativa	true ecommerce	class precision
pred. entretenimiento	8	0	1	2	2	11	1	32.00%
pred. motor de busqueda	0	0	0	0	0	0	0	0.00%
pred. red social	0	0	2	1	0	0	0	66.67%
pred. noticias	2	0	1	2	1	1	3	20.00%
pred. servicios financieros	1	0	0	0	5	0	2	62.50%
pred. informativa	16	3	3	12	4	31	11	38.75%
pred. ecommerce	2	0	0	3	1	1	5	41.67%
class recall	27.59%	0.00%	28.57%	10.00%	38.46%	70.45%	22.73%	
Accuracy								38.41%

Figura 5.38: Red Neuronal Artificial
Fuente: Elaboración propia

no mediante un proceso que optimiza parámetros considerando Sensibilidad de estos y realizando variaciones de forma lineal hasta llegar al óptimo o hasta llegar a un número dado de iteraciones, en este caso el número máximo de iteraciones se fija en 4 por parámetro.

Finalmente la optimización se logra bajo los parámetros Learning rate = 1.0 y $Momentum = 0{,}00001$ (figura 5.39), logrando un $Accuracy = 37{,}68\,\%$ (ver figura 5.40).

ANN setup optimizado	
Ciclos de entrenamiento	550
Tasa de aprendizaje	1.0
Momentum	0.00001

Figura 5.39: Red Neuronal Artificial
Fuente: Elaboración propia

Este resultado es muy poco satisfactorio por la simple razón de que es peor que la linea base, esto ocurre debido a que el algoritmo de optimización ha encontrado un óptimo local bajo, que es el mejor en su vecindad, pero peor que otro óptimo, por ejemplo uno que debiese encontrarse cerca del punto de partida. No se intenta encontrar un mejor óptimo puesto que los resultados entregados por este algoritmo son muy bajos en comparación a los del algoritmo de *NaiveBayes* y el tiempo de procesamiento para este algoritmo supera los 7 días considerando las condiciones actuales.

A continuación se muestra un resumen de los modelos de minería de datos que se implementan

Red Neuronal Artificial								
	true entretenimiento	true motor de busqueda	true red social	true noticias	true servicios financieros	true informativa	true ecommerce	class precision
pred. entretenimiento	7	1	1	8	0	4	2	30.43%
pred. motor de busqueda	0	0	0	0	0	0	0	0.00%
pred. red social	1	0	0	0	2	2	0	0.00%
pred. noticias	6	0	0	7	0	0	8	33.33%
pred. servicios financieros	0	1	1	1	0	0	0	0.00%
pred. informativa	13	1	5	3	11	38	12	45.78%
pred. ecommerce	2	0	0	1	0	0	0	0.00%
class recall	24.14%	0.00%	0.00%	35.00%	0.00%	86.30%	0.00%	
Accuracy								37.68%

Figura 5.40: Red Neuronal Artificial
Fuente: Elaboración propia

bajo las condiciones que se describen en esta sección.

Resumen modelamiento considerando texto visible					
Algoritmo	Naive Bayes	SVM	Random Forest	Decision Tree	ANN
Accuracy inicial	73.60%	33.33%	31.88%	55.80%	38.41%
Accuracy optimizado	**77.74%**	**35.51%**	**36.96%**	**57.25%**	**37.68%**

Figura 5.41: Resumen algoritmos considerando texto visible como variables de decisión
Fuente: Elaboración propia

5.3.2. Minería de datos y Seguridad web como variable de decisión

En el capítulo anterior, en conjunto con el marco teórico se describe que se entiende por seguridad web, como se obtienen los certificados de seguridad y que significa cada letra según el ranking de SSL LABS. En este capítulo se quiere probar la premisa que expone el autor del rating de SSL LABS que dice: .*El puntaje que un sitio web debiese tener no es un número fijo, debido a que diferentes sitios web tienen distintas necesidades, el servidor SSL debe configurarse de acuerdo a esas y estar preparado para las amenazas que implique su entorno, si bien la labor de SSL LABS no es decir como se debe configurar un servidor SSL, SSL LABS provee consejos sobre que no hacer"* [45], la importancia de esto radica en que se implica una correlación entre el tipo de seguridad que implementa una página web y la función que esta página desea desarrollar (categorización por función), por lo tanto la etapa final de este trabajo en lo que respecta a optimización del modelo de clasificación de páginas web es comprobar si incluir la seguridad web como variable de decisión mejora el *Accuracy* y por consiguiente es relevante.

5.3.2.1. Algoritmos de minería de datos sobre juego de variables que considera seguridad web

El vector de características en la iteración anterior está compuesto por 39617 atributos, por lo tanto agregar una nueva variable no hace una diferencia significativa en términos de procesamiento, es por eso que se decide utilizar el mismo preprocesamiento que se utilizó para el ejercicio anterior (ver figura 5.20).

Es importante recordar que el juego de datos cuenta con 138 observaciones repartidas en siete clases (Noticias, Entretenimiento, Red social, Noticias, Servicios financieros, Motor de búsqueda, E-commerce), se realiza TF-IDF para la obtención de información relevante a partir del contenido visible, en cuyo proceso se decide no utilizar Steming debido a que se obtuvieron resultados poco favorables, probablemente debido a la presencia de más de un idioma en los textos y finalmente la evaluación se realiza mediante validación cruzada, en la cual se utilizan diez particiones.

La linea base para este estudio está dada por los resultados del algoritmo que obtuvo el mejor desempeño considerando el juego de variables que aún no utilizaba seguridad web, es decir la linea base está dada por *NaiveBayes* cuyo *Accuracy* fue de 77,54 %.

Además debido a que la mayor parte del modelo está explicado por el juego de variables sin considerar seguridad web, solo se realizará el estudio con los modelos que obtuvieron resultados satisfactorios, los cuales corresponden a los algoritmos de Naive Bayes y decision Tree.

1. ***Naive Bayes***

   El algoritmo *NaiveBayes* ha demostrado ser un algoritmo simple con muy buenos resultados durante este trabajo, se considera la linea base para este clasificador con pruning en el proceso TF-IDF que considera eliminar valores bajo el 5 % de frecuencia y sobre un 95 % de frecuencia con respecto al total del ranking y sin considerar corrección de *Laplace*.

   La primera aproximación de *NaiveBayes* considerando seguridad web logra un $Accuracy = 75{,}91\,\%$ (ver figura 5.42), que si bien es un resultado con menor *Accuracy* que la linea base que se desea superar, es un resultado superior al del mismo algoritmo sin considerar seguridad web que entrega un $Accuracy = 75{,}20\,\%$.

   El algoritmo se optimiza considerando la corrección de *Laplace* y el proceso de pruning dentro de TF-IDF, el segundo parámetro a considerar no logra mejores resultado al realizar análisis de Sensibilidad por lo que se mantienen los valores, con respecto a la corrección de *Laplace*, fue un parámetro cuyo impacto si fue considerable al momento de optimizar el modelo.

   El desempeño del modelo bajo las consideraciones anteriores logró un $Accuracy = 78{,}67\,\%$ (ver figura 5.43), resultado superior a todas las aproximaciones hasta el momento.

2. ***Decision Tree Classificator***

Naive Bayes								
	true entretenimiento	true motor de busqueda	true red social	true noticias	true servicios financieros	true informativa	true ecommerce	class precision
pred. entretenimiento	24	1	2	0	0	1	2	80.00%
pred. motor de busqueda	0	0	0	0	0	0	0	0.00%
pred. red social	1	1	4	0	0	0	0	66.67%
pred. noticias	0	0	0	16	1	5	1	69.57%
pred. servicios financieros	1	0	0	0	10	1	1	76.92%
pred. informativa	1	1	0	5	1	33	2	76.74%
pred. ecommerce	1	0	0	1	0	3	17	77.27%
class recall	85.71%	0.00%	66.67%	72.73%	83.33%	76.74%	73.91%	
Accuracy								75.91%

Figura 5.42: Naive Bayes considerando seguridad web
Fuente: Elaboración propia

Naive Bayes								
	true entretenimiento	true motor de busqueda	true red social	true noticias	true servicios financieros	true informativa	true ecommerce	class precision
pred. entretenimiento	24	0	2	0	0	2	2	80.00%
pred. motor de busqueda	3	0	1	1	2	1	1	0.00%
pred. red social	0	2	4	0	0	0	0	66.67%
pred. noticias	1	0	0	20	1	6	0	71.43%
pred. servicios financieros	0	1	0	0	10	1	0	83.33%
pred. informativa	1	0	0	1	2	38	1	88.37%
pred. ecommerce	0	0	0	0	0	0	22	100.00%
class recall	82.76%	0.00%	57.14%	90.91%	66.67%	79.17%	84.62%	
Accuracy								78.67%

Figura 5.43: Naive Bayes optimizado considerando seguridad web
Fuente: Elaboración propia

El algoritmo Decision Tree obtuvo el segundo mejor desempeño considerando el vector de características inicial (considerando variables relativas a contenido web) y información relacionada a texto visible logrando un $Accuracy = 57{,}25\,\%$. En esta instancia la linea base para el clasificador se define considerando los aprendizajes de la primera iteración (ver figura 5.44), es decir, considerar como criterio de ganancia *Accuracy* y considerar *pruning* y

prepruning para la creación de los arboles, además para el proceso de TF-IDF también se realiza pruning que considera eliminar valores bajo el 5 % de frecuencia y sobre un 95 % de frecuencia con respecto al total del ranking.

Setup inicial Decision Tree	
Criterio	Accuracy
Máxima profundidad	20
Confianza	0,25
Ganancia mínima	0,1
Mínimo de hojas	2
Mínimo de hojas para dividir	4
Número de prepruning	3
Voting strategy	Confianza
Pruning	True
Pre-pruning	Tue

Figura 5.44: Setup inicial Decision Tree
Fuente: Elaboración propia

El setup inicial considerando la nueva variable logra un $Accuracy = 58{,}67\,\%$, resultado que representa una mejora aún considerando el modelo optimizado de la iteración anterior.

Decision Tree								
	true entretenimiento	true motor de busqueda	true red social	true noticias	true servicios financieros	true informativa	true ecommerce	class precision
pred. entretenimiento	15	2	2	3	2	5	3	46.88%
pred. motor de busqueda	0	0	0	0	0	0	0	0.00%
pred. red social	3	0	3	0	1	1	0	37.50%
pred. noticias	5	0	0	12	1	2	0	60.00%
pred. servicios financieros	0	0	2	1	6	3	1	46.15%
pred. informativa	4	1	0	3	5	36	6	65.45%
pred. ecommerce	2	0	0	3	0	1	16	72.73%
class recall	51.72%	0.00%	42.86%	54.55%	40.00%	75.00%	61.54%	
Accuracy								58.67%

Figura 5.45: Decision Tree
Fuente: Elaboración propia

La optimización del modelo consiste en la variación de parámetros relativos al pruning y pre-pruning del proceso de generación de árboles de decisión, esto se realiza a través de análisis de Sensibilidad conjunto de los parámetros, logrando el óptimo con $confianza = 0{,}375$ en el proceso de *pruning* y con $minimalgain = 0{,}001$ en el proceso de *prepruning* (ver figura 5.46).

El desempeño del modelo bajo las consideraciones anteriores logró un $Accuracy = 59{,}33\,\%$

Setup optimizado Decision Tree	
Número de árboles	10
Criterio	Accuracy
Máxima profundidad	10
Confianza	0.375000025
Ganancia mínima	0,001
Mínimo de hojas	2
Mínimo de hojas para dividir	4
Número de prepruning	3
Voting strategy	Confianza
Pruning	True
Pre-pruning	Tue

Figura 5.46: Decision Tree setup optimizado
Fuente: Elaboración propia

(ver figura 5.47), resultado superior al obtenido por el mismo modelo bajo las mismas condiciones y sin considerar la variable relativa a seguridad web.

Decision Tree optimizado								
	true entretenimiento	true motor de busqueda	true red social	true noticias	true servicios financieros	true informativa	true ecommerce	class precision
pred. entretenimiento	16	2	2	3	2	5	3	48.48%
pred. motor de busqueda	0	0	0	0	0	0	0	0.00%
pred. red social	3	0	3	0	1	1	0	37.50%
pred. noticias	5	0	0	12	1	2	0	60.00%
pred. servicios financieros	0	0	2	1	6	3	1	46.15%
pred. informativa	3	1	0	3	5	36	6	66.67%
pred. ecommerce	2	0	0	3	0	1	16	72.73%
class recall	55.17%	0.00%	42.86%	54.55%	40.00%	75.00%	61.54%	
Accuracy								59.33%

Figura 5.47: Decision Tree optimizado
Fuente: Elaboración propia

A continuación se muestra un resumen de los modelos de minería de datos que se implementan bajo las condiciones que se describen en esta sección (figura 6.5).

Resumen modelamiento considerando seguridad web		
Algoritmo	Naive Bayes	Decision Tree
Accuracy inicial	75.91%	58.67%
Accuracy optimizado	78.67%	59.33%

Figura 5.48: Resumen algoritmos considerando seguridad web
Fuente: Elaboración propia

Capítulo 6

Resultados

Los resultados que se exponen en el presente capítulo no solo tienen relación con el desempeño de los algoritmos implementados si no que este capítulo pretende detallar el grado de cumplimiento de los resultados esperados expuestos en el primer capítulo y en caso de que aplique al caso, el desempeño de este resultado junto con el producto resultante.

6.1. Análisis de resultados esperados

Se analizan los resultados esperados que se definen en el capítulo introductorio, los cuales se listan a continuación:

1. Definir las categorías a considerar de sitios web Chilenos.
2. Definir los parámetros del Vector de Características (Feature Vector).
3. Clasificar páginas web.
4. Validar la hipótesis de investigación.

Es necesario mencionar que todos resultados obtenidos en este trabajo son administrados por WIC y todo el código necesario para producirlos, junto con los resultados que ameriten almacenamiento son guardados en `www.BitBucket.com`. A continuación se muestra como se obtiene cada resultado y los detalles pertinentes en cada caso particular.

6.1.1. R1: Definir las categorías a considerar de sitios web Chilenos

El estudio de categorización web se analiza en el capítulo 3, en el cual no solo se realiza un análisis del estado del arte, es decir de las categorías propuestas por otros autores, si no que el capítulo finaliza con una propuesta original sobre las clases que se pueden encontrar considerando el marco de las observaciones del estudio, la realidad país con respecto a la navegación web y

los objetivos de este trabajo. Se definen siete categorías para clasificar páginas web: E-commerce, Noticias, Motores de búsqueda, Entretenimiento, Información, Servicios financieros, Red social.

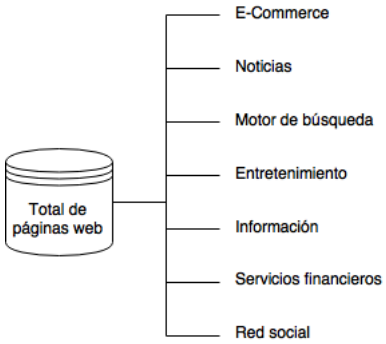


Figura 6.1: Categorización web
Fuente: Elaboración propia

6.1.2. R2: Definir los parámetros del Vector de Características (Feature Vector)

El vector de características corresponde al juego de variables a utilizar por los algoritmos de minería de datos al momento de realizar la clasificación de páginas web, la creación de este vector de características se realiza en tres etapas con respecto a como se obtienen los distintos parámetros.

La primera parte del vector de características viene dada por contenido web, es decir, por objetos extraíbles directamente del contenido HTML, dentro de estos objetos podemos encontrar formularios de ingreso, cantidad de imágenes, caja de búsqueda ("Search box"), entre otros. La primera parte del vector de características consiste en 21 variables en total y su creación puede se encuentra detallada en el capítulo 4.

La segunda parte del vector de características tiene relación con el análisis de todo el texto visible que se encuentra dentro de una página web y las variables corresponden a bi-gramas creados a partir de las palabras de dicho texto. Esta segunda parte del vector de características agrega 39600 variables, llegando a un total de 39617 variables considerando la primera parte.

La tercera parte consiste en la adición de una variable nominal que corresponde a una letra que otorga SSL Labs[1] a un sitio web según la seguridad que haya sido implementada en este, la variable nominal tiene seis valores posibles: A, B, C, D, E, F. La variable nominal se descompone en variables binarias según cada valor que puede tomar la variable nominal, por lo tanto esta parte agrega seis variables al modelo general, llegando a un total de 39623 variables.

[1]La letra es otorgada mediante una aplicación web de su sitio web `www.ssllabs.com/ssltest`

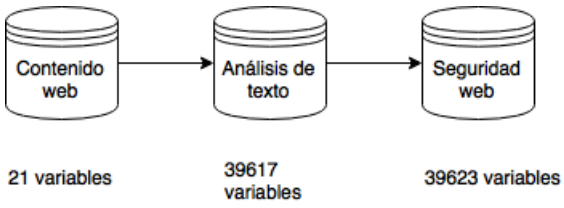


Figura 6.2: Creación del vector de características
Fuente: Elaboración propia

6.1.3. R3: Clasificar páginas web

El presente resultado es considerado el más importante, puesto que habla directamente del desempeño de este trabajo, este resultado permite analizar con cuanta exactitud fue posible clasificar páginas web utilizando como insumo solo URL.

Para la clasificación de páginas web se utilizó un juego de datos de 138 observaciones, distribuido en siete clases[2] y como se expone en el resultado esperado anterior, un juego de variables incremental. Sobre cada juego de variables (considerando que son acumulativos) se aplican variados algoritmos de minería de datos de clasificación todos bajo un enfoque de aprendizaje supervisado y , lo que requiere un etiquetado manual de las clases en el juego de datos, es necesario mencionar que debido a que este estudio no realiza clasificación binaria, es una clasificación entre siete clases, el caso aleatorio considera un accuracy de 14.28 %, el detalle del diseño e implementación del proceso que finaliza con los resultados expuestos a continuación se detalla en el capítulo 5.
A continuación se realiza un análisis considerando las distintas etapas del vector de características.

6.1.3.1. Juego de variables: Contenido web

La primera aproximación para realizar la clasificación de páginas web se realiza considerando únicamente el contenido extraíble a partir del código HTML de dichas páginas web, en este escenario el mejor desempeño fue alcanzado por la red neuronal artificial (ANN), alcanzando un accuracy de 58.78 %

Resumen modelamiento según contenido HTML					
Algoritmo	Naive Bayes	SVM	Random Forest	Decision Tree	ANN
Accuracy Inicial	52,42%	37,90%	45,16%	42,74%	52,67%
Accuracy Optimizado	**56,36%**	**41,94%**	**56,45%**	**52,42%**	**58,78%**

Figura 6.3: Resumen modelos
Fuente: Elaboración propia

[2]El juego de datos no se encuentra balanceado por clases, si no que sigue la tendencia de las páginas web más visitadas por Chilenos

6.1.3.2. Juego de variables: Contenido web

La segunda aproximación al problema considera el juego de variables propuesto en la primera aproximación y adicionalmente se utiliza como variables de decisión todo el texto visible dentro de una página web (se crean variables a través de bi-gramas los cuales son producto del proceso TF-IDF). Bajo este escenario el desempeña aumenta considerablemente, el cual llega a su máximo en el algoritmo de Naive Bayes logrando un *Accuracy = 77.74 %* (ver figura 6.4).

Resumen modelamiento considerando texto visible					
Algoritmo	Naive Bayes	SVM	Random Forest	Decision Tree	ANN
Accuracy inicial	73.60%	33.33%	31.88%	55.80%	38.41%
Accuracy optimizado	**77.74%**	**35.51%**	**36.96%**	**57.25%**	**37.68%**

Figura 6.4: Resumen algoritmos considerando texto visible como variables de decisión
Fuente: Elaboración propia

6.1.3.3. Juego de variables: Seguridad web

La última parte del juego de variables considera la seguridad web como variable de decisión, que en la práctica corresponde a la nota asignada por SSL Labs a un sitio web que es un reflejo de la seguridad implementada en sus servidores con respecto a la compatibilidad de protocolos, intercambio de llaves e intensidad de cifrado. Bajo este escenario el desempeño aumenta en los dos modelos considerados, por lo que se concluye que es una variable contribuyente en cuanto a clasificación web, lo que logra un desempeño de 78.67 % (ver figura 6.5).

Resumen modelamiento considerando seguridad web		
Algoritmo	Naive Bayes	Decision Tree
Accuracy inicial	75.91%	58.67%
Accuracy optimizado	78.67%	59.33%

Figura 6.5: Resumen algoritmos considerando seguridad web
Fuente: Elaboración propia

Los resultados recién expuestos si bien dan indicios sobre el desempeño del modelo, no entregan información sobre que tan bien se está clasificando considerando la investigación actual, es por eso que es necesario realizar comparaciones contra el estado del arte en la clasificación web.

Es muy poco probable encontrar un experimento de otro investigador que cumpla con todas las características necesarias para realizar una comparación representativa, sin embargo se pueden extrapolar datos y resultados para tener indicios sobre la importancia de los resultados.

Se utiliza como linea base o punto de comparación cuatro experimentos, primero se contrasta contra el trabajo de Rajalakshmi y Aravindan en *Naive Bayes approach for website classification*[64], el cual consiste en utilizar el clasificador de Naive Bayes para clasificar sitios web (no confundir con páginas web) basado solamente en URL, es decir, no utiliza como insumo el contenido del sitio, solo el nombre de la dirección web. esta clasificación se realiza en torno a 9 clases(ver figura 6.6) y al igual que el experimento del presente trabajo utiliza validación cruzada para obtener indicadores de desempeño, sin embargo considera F-measure (también conocido como F1-Score)

como su principal métrica de desempeño, la cual no es directamente comparable con Accuracy, es por eso que se calcula dicha métrica de forma de poder realizar una comparación considerando las circunstancias.

Category	Number of URLs	Training Set	Testing Set
Arts	227337	204597	22740
Business	227000	204300	22700
Computers	109201	98390	10921
Games	51207	46080	5127
Health	57515	51750	5755
Home	25368	22824	2544
News	8235	7407	828
Recreation	94565	85104	9461
Reference	55521	49967	5554
Total	8,55,939	7,70,309	85,630

Figura 6.6: Clases del experimento
Fuente: Naive Bayes approach for website classification
R Rajalakshmi, C Aravindan

F-measure es una medida que considera tanto *Precision* como *Recall*, que en su versión estándar ambas tienen la misma importancia, esta métrica se define como:

$$F-measure = 2 * \frac{precision * recall}{precision + recall}$$

Sin embargo debido a que es un problema de clases múltiples, tanto la *precision* como el recall corresponden a sumas ponderadas de cada clase:

$$Precision = \frac{TruePositive}{Positive} = \sum_{i=1}^{N} \frac{TruePositive_i}{\sum_{j=1}^{N} Positive_j} * w_i, w_i = \frac{\sum_{i=1}^{N} True_i}{\sum_{i=1}^{N}\sum_{j=1}^{N} True_{ij}}$$

$$Recall = \frac{TruePositive}{True} = \sum_{i=1}^{N} \frac{TruePositive_i}{\sum_{j=1}^{N} True j} * w_i, w_i = \frac{\sum_{i=1}^{N} True_i}{\sum_{i=1}^{N}\sum_{j=1}^{N} True_{ij}}$$

Es necesario destacar que *F-measure* no es una fracción de un todo, sino que corresponde a un radio entre casos favorables con respecto a distintos subconjuntos que resultan en una métrica para medir el desempeño de un modelo considerando la disparidad que puede existir entre *precision* y *recall*.

Debido a su construcción este indicador logra su máximo en uno y su mínimo en cero. El trabajo de Rajalakshmi y Aravindan logra un F-measure de 0.76 en su mejor iteración, en comparación a 0.808 que logra el algoritmo de este trabajo, lo cual no necesariamente expresa que el resultado de este trabajo es mejor, puesto que como se dijo anteriormente, las condiciones son substancialmente

distintas, pero sirve de referencia de que el resultado se encuentra cerca del desempeño que debiese tener un trabajo de nivel similar.

El segundo trabajo que se utiliza como referencia corresponde a *Fast webpage classification using URL features*[65] que consiste en realizar clasificación de páginas web utilizando como insumo el actual estado del arte tanto en comprensión de textos como de métodos basados en links, sin embargo nuevamente solamente basándose en lo que entrega el texto de la URL. Este trabajo considera solo URL del tópico de educativo y las clasifica en "student", "faculty", "course" y "project", logrando un accuracy de 76 % utilizando el algoritmo SVM bajo validación cruzada, la cual es menor que el 78.67 % por lo que se concluye lo mismo que en el caso anterior, el trabajo se encuentra en el rango aceptable de desempeño.

El tercer trabajo llamado Joint Web-Feature (JFEAT): A Novel Web Page Classification Framework[66], realiza una clasificación que considera URL, web page title información extraída de metadata y categorías definidas por los mismos usuarios. Este trabajo considera seis categorías (ver tabla 6.7). Este trabajo logra un F-measure de 78.6 % utilizando el algoritmo de Maximum Entropy bajo validación cruzada, este resultado también es inferior al obtenido en el trabajo presente.

Category	Number of Web Pages
BusinessEconomy	79
Entertainment	38
Government	43
Health	88
News	40
Sports	49

Figura 6.7: Clases del experimento
Fuente: Joint Web-Feature (JFEAT): A Novel Web Page Classification Framework

El cuarto trabajo llamado Web page genre classification[67] considera para la clasificación no solo el contenido dentro de una página web, sino que también URLs, HTML tags, Java scripts, y VB scripts. El trabajo realiza la clasificación dentro de cuatro categorías: "Online shopping", "Discussion forum", "University homepage" y "Frecuently asked Questions". Lo cual logra un accuracy de 93 % utilizando un algoritmo que consiste en la asignación de pesos según la frecuencia de una variable dentro de la categoría en cuestión, si bien este resultado es muy alto no se logra bajo validación cruzada por lo que podría ser cuestionable.

Se concluye que el desempeño de este trabajo esta dentro de los parámetros aceptables, obtuvo mejores resultados que la mayoría de los trabajos de vanguardia en clasificación web considerando condiciones similares pero datos y categorías distintas.

Es necesario mencionar que el modelo que clasifica de mejor manera páginas web no fue capaz de clasificar ninguna observación perteneciente a la clase motor de búsqueda, esto se puede explicar por la baja cantidad de observaciones (3 de 138) y por otro lado la clase *noticias* fue la clase que tuvo la mayor cantidad de observaciones clasificadas correctamente logrando un *Recall = 90.91 %*, sin embargo su *Precision = 71.43 %* no fue bueno comparativamente con las demás clases, esto quiere decir que el modelo clasifica un gran número de observaciones en dicha clase.

La clase E-Commerce al parecer es la que tiene las variables más distintivas puesto que el

Resumen de clasificación web									
	Variables a considerar								
Titulo	URL	Contenido HTML	Texto visible	Seguridad web	Algoritmo	Accuracy	F-measure	Validación	Clases
Desarrollo de módulo para clasificar páginas web con respecto a características excluyentes utilizando como insumo URL	Si	Si	Si	Si	Naive Bayes	78.67%	0.808	cruzada	7
Naive bayes approach for website classification	Si	No	No	No	Naive Bayes	-	0.76	cruzada	9
Fast webpage classification using URL features	Si	No	No	No	SVM	76%	0.525	cruzada	4
Web page genre classification	Si	Si	No	No	Algorimo propio	93%	0.91	-	4
Joint web-feature (JFEAT): A novel web page classification framework	Si	Si	No	No	ME	-	0.786	cruzada	6

Figura 6.8: Resumen clasificación web
Fuente: Elaboración propia

modelo fue capaz de clasificar con una *Precision de 100%*, sin embargo su *Recall* es de 84.62% que si bien es alta, no es perfecta, lo que se puede otorgar a que las variables excluyentes de páginas web de E-Commerce (carro de compra por ejemplo) no se encuentran en todas las observaciones (páginas de cupones: `www.groupon.cl` o listing: `www.mercadolibre.cl`).

Es importante destacar que no se puede concluir que las clases con mayor cantidad de observaciones se clasifican con mayor desempeño puesto que las dos clases con mayor cantidad de observaciones: Informativas (48 observaciones) y Entretenimiento (29 observaciones) obtuvieron resultados sobre el promedio pero no destacables.

6.1.3.4. Variables clave

El cálculo de la importancia de las variables al momento de clasificar no fue definido como un objetivo específico ni como un resultado esperado, sin embargo entrega información sobre cuales pueden ser las características clave que definen el paso entre una clase de página web y otra, además permite entender de mejor manera el comportamiento del consumir actual, puesto que nos dice cuales son las características clave para que un sitio web pueda ser considerado de una cierta clase aun cuando el mismo usuario podría no saberlo.

En la figura C.1 se muestran los pesos de las 20 variables más importantes que considera el juego de variables completo, este análisis se realiza bajo cuatro modelo para el calculo de pesos

1. Matriz de correlación.
2. Chi cuadrado
3. Según correlación.
4. Según PCA.

Los cuatro son métodos comunes para este tipo de análisis por lo que no se realizará un análisis exhaustivo como se ha realizado a lo largo de este trabajo, sin embargo es necesario mencionar una distinción entre matriz de correlación y cálculo de pesos según correlación, la primera realiza un análisis de la correlación entre las variables y el resultado final (pesos) se basa en dichos resultados,

en el segundo método se realiza el análisis de cada atributo por separado y se calcula la correlación de este contra el valor de la etiqueta y en base a este resultado se asigna el peso de la variable.

El resultado de este análisis se realiza de forma cualitativo basado en técnicas cuantitativas, es decir, no se realiza un ranking considerando un promedio ponderado entre los análisis recién expuestos, debido que sería sesgado en la misma medida en que el análisis que se presentará a continuación.

Se decide exponer las variables más importantes al momento de clasificar una página web a todas las variables que se repiten a lo largo de los cuatro métodos de asignación de pesos y figuran entre las primeras 20.

1. Carro de compras
2. Price Tag
3. Input Submit
4. Password
5. Security Grade
6. Image number
7. Dominio

Muchas de ellas fueron intuitivas por su pertenencia y exclusividad de solo un tipo de páginas web como por ejemplo *Carro de compra* o *dominio* (tv en entretenimiento) o con menor exclusividad pero con gran poder como el *price tag* en E-commerce o Servicios financieros, esto entrega información muy valiosa de que la web posee características que no son intuitivas y el modo en que se reconoce un tipo de página web podría no ser conocido aún por el propio usuario que clasifica.

6.1.4. R5: Validar la hipótesis de investigación

La hipótesis de investigación es: “Es posible clasificar páginas web mediante su contenido”, esta hipótesis se válida considerando que el modelo fue capaz de clasificar páginas web con un accuracy y F-measure similar al que obtienen los trabajos de investigación de vanguardia actualmente, además estos resultados son particularmente confiables puesto que utiliza validación cruzada con el objetivo de disminuir el sobreajuste de los datos de la mejor manera posible.

Capítulo 7

Conclusiones

En el presente capítulo se dan a conocer las conclusiones de este trabajo tanto de los resultados propios de este trabajo como del proceso necesario para llegar a ellos, además se realiza un análisis sobre posibles mejoras que pueden ser implementadas en el futuro, distintos enfoques y aplicaciones no relacionadas con el propósito de esta investigación.

7.1. Conclusiones Generales

El presente trabajo de título se ha realizado en el marco del proyecto AKORI, el que une la investigación del área biológica con las herramientas y el conocimiento que puede aportar la ingeniería. En los últimos años AKORI ha desarrollado variados temas de investigación relativos a la identificación de objetos web y la utilización de variables fisiológicas para estudiar el comportamiento en la web, lo que ha resultado en un ambicioso proyecto que pretende predecir mediante técnicas de minería de datos tanto la fijación ocular como la dilatación pupilar sobre una página web dada.

El objetivo general de este trabajo es desarrollar un módulo que sea capaz de clasificar páginas web en base a su contenido web, lo que produce tres potenciales beneficios para el proyecto AKORI. Primero, debido a la clasificación se generan clases, cuyos elementos son muy distintos entre clases y similares dentro de la misma clase, lo que ayuda al momento de entrenar el modelo de AKORI, mejorando su desempeño y poder predictivo, segundo permite personalizar tanto herramientas como vistas de resultados de AKORI para los distintos tipos de clientes (según clase de página web) y tercero entrega una lista de objetos web que se utilizan como variables de entrada para el modelo.

Para cumplir el objetivo propuesto fue necesario realizar un profundo análisis del estado del arte en los temas relacionados, cuya primera conclusión fue que no existe una forma única de categorizar la web, es decir, se pueden definir clases muy distintas para clasificar la web dependiendo del autor. Se definen siete clases para este proyecto: *Entretenimiento, Noticias, Información, E-Commerce, Servicios Financieros, Red Social, Motor de búsqueda*, siempre considerando maximizar la varianza entre clases y minimizar la varianza intra clase y se utiliza un juego de datos de

138 observaciones compuesto por las páginas web más visitadas por Chilenos según Alexa.

Con respecto al estado del arte también se concluye que existe muy poca investigación en la clasificación que considera contenido, la basta mayoría de la investigación se centra solo en la utilización del texto de la URL para clasificar y por lo general no se utilizan páginas aleatorias para clasificar, si no un tópico de la web muy acotado en base a los que se definen las clases.

El modelo utiliza como variables de entrada un vector de características que se compone por tres partes. La primera son variables relacionadas al contenido HTML, en su mayoría son producto del análisis de tags HTML, la segunda relacionadas con análisis del texto visibles, corresponden a bi-gramas resultado de TF-IDF y la tercera con respecto a la seguridad web, la cual se representa con una letra entre la A y F.

Se realiza un análisis de pesos (ponderadores) para analizar cuales son las variables más importantes de este trabajo. Para esta tarea se implementaron cuatro modelos: *Chi cuadrado, Pesos por correlación, Pesos por PCA y matriz de correlación* y se concluye que las variables más importantes son: *Nota de seguridad, Carro de compras, Price tag, Input submit, password, image number y dominio*, de este análisis se desprende que si bien era esperable encontrar variables exclusivas de clases como *Carro de compras* en E-Commerce o *Dominio = tv* en Entretenimiento, existen variables menos intuitivas que tienen gran poder de predicción como *Input submit y password* que son parte de los formularios de acceso y la variable relativa a seguridad web, que se considera la originalidad de este proyecto.

En el proceso de minería de datos, que se desarrolla bajo la metodología KDD, se implementan cinco algoritmos de clasificación: *Support Vector Machine, Naive Bayes, Decision Tree, Random Forest y Artificial Neural Network*, de los cuales *Naive Bayes* y *Decision Tree* obtienen los mejores desempeños con respecto a *accuracy*, logrando 78.76 % y 59.33 % respectivamente. El resultado es satisfactorio considerando que no se pudieron encontrar trabajos de clasificación web con mejores resultados bajo validación cruzada.

Al analizar detalladamente los resultados del mejor modelo, se aprecian ciertas particularidades, primero que el modelo no fue capaz de clasificar ninguna observación relativa a motores de búsqueda, sin embargo dicha clase solo posee 3 observaciones de un total de 138 por lo que ese resultado es explicable por la falta de información y segundo punto importante es que cada observación clasificada como E-Commerce fue clasificada correctamente ($precision = 100\,\%$), lo que puede ser explicado debido a que en el análisis de pesos, las variables más importantes tienen relación con E-Commerce (todas tienen importancia en E-Commerce, a excepción de dominio). Notar que E-Commerce no logra $Recall = 100\,\%$, lo que se explica debido a que páginas de E-Commerce no tienen las variables diferenciadoras como *Carro de compras* (Ejemplo: páginas de cupones: `www.groupon.cl` o listing: `www.mercadolibre.cl`).

A modo de conclusión final se termina el trabajo con resultados satisfactorios, se logran todos los resultados esperados, se valida la hipótesis de investigación y se logra un desempeño de clasificación web comparable al obtenido por los trabajos de vanguardia en el rubro.

7.2. Trabajo futuro y recomendaciones

A partir del trabajo realizado, de los resultados obtenidos y de las conclusiones se proponen las siguientes mejoras para el módulo que clasifica páginas web utilizando contenido web:

1. Realizar estudios con respecto a los potenciales clientes de AKORI con el objetivo de caracterizarlos y categorizarlos y así de reducir el número de clases y si es posible, generar clases aún más excluyentes con el objetivo de mejorar el desempeño del modelo y entregar un mejor servicio al WIC.

2. En base al trabajo actual, construir un modelo que no solo le asigne una clase a la página web a analizar, si no que entregue como resultado una probabilidad de pertenencia a cada clase con el objetivo de proveer recomendaciones para los usuarios del servicio. Si bien las recomendaciones serian de carácter cualitativo, podrían dar fuertes indicios de que una página web podría no estar transmitiendo la imagen del servicio que desea ofrecer y por otro lado describir y analizar la hibridez de una página web.

3. La identificación de objetos web ha sido tema de investigación desde hace muchos años en el WIC, lo que ha resultado en una metodología que permite obtener resultados que aún deben ser procesados manualmente para su posterior utilización, se propone desarrollar un sistema que permita la identificación de todos los objetos web dentro de una página web de forma automática, y de esta forma ser capaz de analizar todos los objetos que componen un sitio web de forma rápida y eficiente. Esta propuesta beneficiaria también a toda la investigación fisiológica de AKORI, con resultados más rápidos y eliminando el error potencial humano en esta etapa.

4. Para mejorar el desempeño del modelo es posible desarrollar nuevamente el módulo considerando todo el contenido *Javascript* de las páginas web dinámicas con el objetivo de parametrizar una página web de mejor forma, con información más específica y posiblemente encontrar otras variables excluyentes entre tipos de páginas web.

5. Aumentar considerablemente el juego de datos con el objetivo de no solo lograr una muestra más representativa para cada clase si no que el modelo aprenda más características de cada clase (en especial por texto) y las características comunes entre algunas observaciones de distintas clases tengan menos peso.

6. Considerar una cuarta parte del vector de características relacionada con estructura de un sitio web, es decir, que se estudie el mapeo de hipervínculos dentro un sitio, como difiere este entre sitios web de una clase y otra y si tiene poder predictivo al momento de clasificar páginas web.

7. Estudiar el impacto de la adición de un set de variables que estudie el contenido de las imágenes dentro de un sitio, similar al servicio que ofrece Google a través de su aplicación *Google Cloud Vision*[1], la que categoriza imágenes en una alta gama de contenidos desde medios de transporte a animales.

[1]Disponible en `https://cloud.google.com/vision/`

8. Estudiar el impacto de implementar una variable de distancia entre una página preestablecida en una clase de páginas web (observación modelo) y una observación cualquiera, la distancia se define según la distancia a la que la búsqueda orgánica de un motor de búsqueda los posiciona.

9. Estudiar si existe relación entre las variables relacionadas con objetos web más importantes según clasificación web y los objetos que reciben más atención por parte del comportamiento ocular, tanto por fijación ocular como por dilatación pupilar.

Bibliografía

[1] I. W. Stats. (2016). World internet users statistics and 2016 world populations stats. consulta: 25 Octubre 2016, dirección: `http://www.internetworldstats.com/stats.htm`.

[2] InternetWorldStats. (2016). South american internet users statistics. consulta: 25 Octubre 2016, dirección: `http://www.internetworldstats.com/stats15.htm`.

[3] R. Benbunan-Fich, "Using protocol analysis to evaluate the usability of a commercial web site", *Information & management*, vol. 39, n.º 2, págs. 151-163, 2001.

[4] C. Liu y K. P. Arnett, "Exploring the factors associated with web site success in the context of electronic commerce", *Information & management*, vol. 38, n.º 1, págs. 23-33, 2000.

[5] R. V. McCarthy y J. E. Aronson, "Activating consumer response: A model for web site design strategy", *Journal of Computer information systems*, vol. 41, n.º 2, págs. 2-8, 2001.

[6] WorldWideWebSize. (1999). The size of the world wide web (the internet), dirección: `http://www.worldwidewebsize.com/` (visitado 25 de oct. de 2016).

[7] X. Qi y B. D. Davison, "Web page classification: Features and algorithms", *ACM Computing Surveys (CSUR)*, vol. 41, n.º 2, pág. 12, 2009.

[8] J. D. Velásquez, R. Weber, H. Yasuda y T. Aoki, "A methodology to find web site keywords", en *E-Technology, e-Commerce and e-Service, 2004. EEE'04. 2004 IEEE International Conference on*, IEEE, 2004, págs. 285-292.

[9] L. E. Dujovne y J. D. Velásquez, "Design and implementation of a methodology for identifying website keyobjects", en *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, Springer, 2009, págs. 301-308.

[10] L. J. González González, "Mejoramiento de una metodología para la identificación de website keyobjects mediante la aplicación de tecnologías eye tracking y algoritmos de web mining", 2011.

[11] C. F. Aracena Cornejo, "Estudio de la relación entre neurodatos, dilatación pupilar y emocionalidad basado en técnicas de minería de datos", 2014.

[12] J. N. Jadue Musalem, "Incidencia de la dilatación pupilar como variable predictiva del comportamiento de los usuarios en una página web antes de tomar una decisión", 2014.

[13] G. A. Slanzi Rodríguez, "Estudio del impacto del uso de electroencefalograma en la identificación de website keyobjects", 2014.

[14] Vox. (2016). How the internet was created. consulta: 25 Octubre 2016, dirección: http://www.vox.com/a/internet-maps.

[15] Statista. (2016). Number of connected devices worldwide. consulta: 25 Octubre 2016, dirección: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/.

[16] webopedia. (2016). What is the internet. consulta: 25 Octubre 2016, dirección: http://www.webopedia.com/TERM/I/Internet.html.

[17] Pewinternet. (2016). World wide web timeline. consulta: 25 Octubre 2016, dirección: http://www.pewinternet.org/2014/03/11/world-wide-web-timeline/.

[18] W3. (2016). Architecture of the world wide web, volume one. consulta: 25 Octubre 2016, dirección: http://www.w3.org/TR/webarch/.

[19] Webcitation. (2016). Website. consulta: 25 Octubre 2016, dirección: http://www.webcitation.org/6EV535JZ1.

[20] Computerhope. (2016). Web page. consulta: 25 Octubre 2016, dirección: http://www.computerhope.com/jargon/w/webpage.htm.

[21] Oracle. (2016). Url. consulta: 25 Octubre 2016, dirección: https://docs.oracle.com/javase/7/docs/api/java/net/URL.html.

[22] C. Castillo, "Effective web crawling", en *ACM SIGIR Forum*, Acm, vol. 39, 2005, págs. 55-56.

[23] E. Vargiu y M. Urru, "Exploiting web scraping in a collaborative filtering-based approach to web advertising", *Artificial Intelligence Research*, vol. 2, n.º 1, p44, 2012.

[24] L. González y J. D. Velásquez, "Una aplicación de herramientas de eye-tracking para analizar las preferencias de contenido de los usuarios de sitios web", *Revista de Ingeniería de Sistemas*, vol. 26, n.º 1, págs. 95-118, 2012.

[25] J. D. Velásquez y L. C. Jain, *Advanced techniques in web intelligence*. Springer, 2010, vol. 311.

[26] IGI Global. (2016). Feature vector. consulta: 25 Octubre 2016, dirección: http://www.igi-global.com/dictionary/feature-vector/10972.

[27] U. Fayyad, G. Piatetsky-Shapiro y P. Smyth, "From data mining to knowledge discovery in databases", *AI magazine*, vol. 17, n.º 3, pág. 37, 1996.

[28] O. Maimon y L. Rokach, *Data mining and knowledge discovery handbook*. Springer, 2005, vol. 2.

[29] R. Jin y Z. Ghahramani, "Learning with multiple labels", en *Advances in neural information processing systems*, 2002, págs. 897-904.

[30] L. Rokach y O. Maimon, "Classification trees", en *Data mining and knowledge discovery handbook*, Springer, 2009, págs. 149-174.

[31] L. Breiman, J. Friedman, C. J. Stone y R. A. Olshen, *Classification and regression trees*. CRC press, 1984.

[32] A. Liaw y M. Wiener, "Classification and regression by randomforest", *R news*, vol. 2, n.º 3, págs. 18-22, 2002.

[33] G. P. Zhang, "Neural networks for data mining", en *Data mining and knowledge discovery handbook*, Springer, 2009, págs. 419-444.

[34] P. Sebastiani, M. M. Abad y M. F. Ramoni, "Bayesian networks", en *Data Mining and Knowledge Discovery Handbook*, Springer, 2005, págs. 193-230.

[35] A. Shmilovici, "Support vector machines", en *Data Mining and Knowledge Discovery Handbook*, Springer, 2005, págs. 257-276.

[36] C. C. Aggarwal y C. Zhai, *Mining text data*. Springer Science & Business Media, 2012.

[37] W. B. Frakes y R. Baeza-Yates, "Information retrieval: Data structures and algorithms", 1992.

[38] R. Baeza-Yates y B. Ribeiro-Neto, *Modern Information Retrieval*, 2nd. USA: Addison-Wesley Publishing Company, 2008, ISBN: 9780321416919.

[39] M. Kearns, "A bound on the error of cross validation using the approximation and estimation rates, with consequences for the training-test split", *Neural Computation*, vol. 9, n.º 5, págs. 1143-1161, 1997.

[40] Charles Elkan. (2011). Evaluation classifiers. consulta: 25 Octubre 2016, dirección: `http://web.archive.org/web/20111218192652/http://cseweb.ucsd.edu/~elkan/250B/classifiereval.pdf`.

[41] H. K. Lee, T. Malkin y E. Nahum, "Cryptographic strength of ssl/tls servers: Current and recent practices", en *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, ACM, 2007, págs. 83-92.

[42] R. Oppliger, *SSL and TLS: Theory and Practice*. Artech House, 2009.

[43] OpenSSL. (). Cryptography and ssl/tls toolkit, dirección: https://www.openssl.org/.

[44] (2006). Tsl extensions: Rfc 4366. consulta: 25 Octubre 2016, dirección: https://www.ietf.org/rfc/rfc4366.txt.

[45] Qualys SSL Labs. (2015). Ssl server rating guide. consulta: 25 Octubre 2016, dirección: www.ssllabs.com/downloads/SSL_Server_Rating_Guide.pdf.

[46] A. Sun, E.-P. Lim y W.-K. Ng, "Web classification using support vector machine", en *Proceedings of the 4th international workshop on Web information and data management*, ACM, 2002, págs. 96-99.

[47] M. Tarafdar y J. Zhang, "Analysis of critical website characteristics: A cross-category study of successful websites", *Journal of Computer Information Systems*, vol. 46, n.º 2, págs. 14-24, 2005.

[48] X. Qi y B. D. Davison, "Web page classification: Features and algorithms", *ACM Computing Surveys (CSUR)*, vol. 41, n.º 2, pág. 12, 2009.

[49] L. W. Han y M Saadat, "Joint web-feature (jfeat): A novel web page classification framework", *Communications of the IBIMA*, 2010.

[50] B. Pan, H. A. Hembrooke, G. K. Gay, L. A. Granka, M. K. Feusner y J. K. Newman, "The determinants of web page viewing behavior: An eye-tracking study", en *Proceedings of the 2004 symposium on Eye tracking research & applications*, ACM, 2004, págs. 147-154.

[51] M. Tarafdar y J. Zhang, "Analysis of critical website characteristics: A cross-category study of successful websites", *Journal of Computer Information Systems*, vol. 46, n.º 2, págs. 14-24, 2005.

[52] A. Broder, "A taxonomy of web search", en *ACM Sigir forum*, ACM, vol. 36, 2002, págs. 3-10.

[53] M.-Y. Kan y H. O. N. Thi, "Fast webpage classification using url features", en *Proceedings of the 14th ACM international conference on Information and knowledge management*, ACM, 2005, págs. 325-326.

[54] S. Dumais y H. Chen, "Hierarchical classification of web content", en *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, 2000, págs. 256-263.

[55] CheckUpDown. (). Error http 403 forbidden (prohibido). consulta: 25 Octubre 2016, dirección: http://www.checkupdown.com/status/E403_es.html.

[56] C. Ramirez D. ; Espinosa. (2011). El cifrado web (ssl/tls). consulta: 25 Octubre 2016, dirección: http://revista.seguridad.unam.mx/numero-10/el-cifrado-web-ssltls.

[57] D. Hills, R. Downs, R Duerr, J. Goldstein, M. Parsons y H. Ramapriyan, "The importance of data set provenance for science", *Eos*, vol. 96, 2015.

[58] H. Buhle J.; Faye. (2016). How do asian and western websites differ, and why? recent findings in experimental psychology implicate basic differences in cognitive processing. consulta: 25 Octubre 2016, dirección: `http://uxpa2016.org/event/how-do-asian-and-western-websites-differ-and-why-recent-findings-experimental-psychology`.

[59] N. Vratonjic, J. Freudiger, V. Bindschaedler y J.-P. Hubaux, "The inconvenient truth about web certificates", en *Economics of information security and privacy iii*, Springer, 2013, págs. 79-117.

[60] I. Rec, "X. 509 information technology–open systems interconnection–the directory: Public-key and attribute certificate frameworks", Technical report, ITU, inf. téc., 2005.

[61] T Mitchell, "Generative and discriminative classifiers: Naive bayes and logistic regression, 2005", *Manuscript available at http://www. cs. cm. edu/˜ tom/NewChapters. html*,

[62] M Bashiri y A. F. Geranmayeh, "Tuning the parameters of an artificial neural network using central composite design and genetic algorithm", *Scientia Iranica*, vol. 18, n.º 6, págs. 1600-1608, 2011.

[63] C. C. Aggarwal y C. Zhai, *Mining text data*. Springer Science & Business Media, 2012.

[64] R Rajalakshmi y C Aravindan, "Naive bayes approach for website classification", en *Information Technology and Mobile Communication*, Springer, 2011, págs. 323-326.

[65] M.-Y. Kan y H. O. N. Thi, "Fast webpage classification using url features", en *Proceedings of the 14th ACM international conference on Information and knowledge management*, ACM, 2005, págs. 325-326.

[66] L. W. Han y M Saadat, "Joint web-feature (jfeat): A novel web page classification framework", *Communications of the IBIMA*, 2010.

[67] G. Chen y B. Choi, "Web page genre classification", en *Proceedings of the 2008 ACM symposium on Applied computing*, ACM, 2008, págs. 2353-2357.

Anexo A

Juego de Variables

A.1. Juego de Variables de contenido HTML

Variable	Tipo de Variable
SUBMIT (INPUT)	BINARIA
PASSWORD	BINARIA
IMAGE	ENTERO
SUBMIT (BUTTON)	BINARIA
GET TIME	BINARIA
CARRO DE COMPRAS	BINARIA
FAQ	BINARIA
PRICE TAG	BINARIA
COPYRIGHT	BINARIA
EMAIL INFO	BINARIA
TEL NUM	BINARIA
META DESCRIPTION	NOMINAL
PANEL SIZE	NOMINAL
PERSONAS | EMPRESAS	BINARIA
SBIF	BINARIA
CINTA DINÁMICA	BINARIA
DOMINIO	NOMINAL
CERTIFICADO SSL	NOMINAL

Figura A.1: Juego Variables de contenido HTML
Fuente: Elaboración propia

A.2. Juego de Variables de texto

Attirubte name	Frecuencia total	Entretenimiento	Motor de	Red social	Noticias	Servicios financier	Informativas	Ecommerce
abigaile_johnson	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
abogada	2.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0
abogada_anita	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
abogada_manuela	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
abogado	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
abogado_escrupulos	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
abogados	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
abogados_acceder	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
abono	2.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0
abono_inmediato	2.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0
abordado	2.0	0.0	0.0	0.0	0.0	0.0	2.0	0.0
abordado_tercera	2.0	0.0	0.0	0.0	0.0	0.0	2.0	0.0
abordaje	4.0	0.0	0.0	0.0	0.0	0.0	4.0	0.0
abordaje_ministro	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
abordaje_violencia	3.0	0.0	0.0	0.0	0.0	0.0	3.0	0.0
abordar	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
abordar_rumbo	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
abortiva	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
abortiva_senalo	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
abortivo	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
abortivo_explicacion	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
aborto	3.0	0.0	0.0	0.0	0.0	0.0	3.0	0.0
aborto_hijo	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
aborto_inconstitucional	2.0	0.0	0.0	0.0	0.0	0.0	2.0	0.0
about	3.0	1.0	0.0	0.0	0.0	0.0	0.0	2.0
about_amazon	2.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0
about_doggystyle	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0

Figura A.2: Juego Variables de texto
Fuente: Elaboración propia

A.3. Variable de seguridad

URL	[Security Grade]
https://www.youtube.com	[A]
https://www.google.cl	[A]
https://www.facebook.com	[B]
http://www.emol.com	S/C
http://www.biobiochile.cl	S/C
http://www.lun.com	S/C
http://www.bing.com	[A]
https://www.bancoestado.cl	[C]
https://twitter.com/	[A]
http://home.sii.cl	[A]
http://www.mercadolibre.cl	[B]
https://www.instagram.com	[B]
http://www.falabella.com/falabel	S/C
http://www.emol.com	S/C
https://www.netflix.com/	[B]
https://www.linkedin.com	[A]
http://www.santander.cl	[B]
http://simple.ripley.cl	S/C
https://www.transbank.cl/public/	[B]
http://www.sodimac.cl/sodimac-c	S/C
https://plp.cl	[A]
http://www.bci.cl/	[A]
http://www.jumbo.cl/FO/LogonFc	[F]
http://www.estrellaiquique.cl/im	S/C
http://www.udd.cl	[C]
http://www.mercadolibre.cl	[B]
http://www.yapo.cl	[F]
http://larepublica.pe/	S/C
http://rpp.pe/	S/C
http://www.marca.com/	S/C
http://www.uni.edu.pe/	S/C
http://ojo.pe/	S/C
http://www.youporn.com/	[C]
http://www.udp.cl/	S/C
http://www.udd.cl/	[C]
http://store.steampowered.com/	[A]
https://www.ashleymadison.com/	[A]

Figura A.3: Variable de seguridad
Fuente: Elaboración propia

Anexo B

Reducción de dimensionalidad

Component	Standar Dev	Proportion of Var	Cumulative Var
PC 1	62.947	999	999
PC 2	842	0	999
PC 3	698	0	999
PC 4	606	0	999
PC 5	580	0	1.000
PC 6	547	0	1.000
PC 7	531	0	1.000
PC 8	492	0	1.000
PC 9	454	0	1.000
PC 10	437	0	1.000
PC 11	362	0	1.000
PC 12	349	0	1.000
PC 13	268	0	1.000
PC 14	242	0	1.000
PC 15	214	0	1.000
PC 16	180	0	1.000
...	...	...	...
PC 2074	?	0	1.000
PC 2075	?	0	1.000
PC 2076	?	0	1.000
PC 2077	?	0	1.000
PC 2078	?	0	1.000
PC 2079	?	0	1.000
PC 2080	?	0	1.000
PC 2081	?	0	1.000
PC 2082	?	0	1.000
PC 2083	?	0	1.000
PC 2084	?	0	1.000
PC 2085	?	0	1.000
PC 2086	?	0	1.000
PC 2087	?	0	1.000
PC 2088	?	0	1.000
PC 2089	?	0	1.000
PC 2090	?	0	1.000
PC 2091	?	0	1.000
PC 2092	?	0	1.000

Figura B.1: Atributos PCA
Fuente: Elaboración propia

Anexo C

Análisis de pesos

Peso de variables								
	Correlation matrix		Chi squared		Weight by correlation		Weight by PCA	
Ranking	Variable	Peso (Weight)	Variable	Peso (Weight)	Variable	Peso (Weight)	Variable	Peso (Weight)
1	carro compras	1	Dominio	2.329	[Security Grade]	0,439040983	image	99,99731282
2	[input,submit	0,767162647	contactos	1.240	price	0,333238953	carro compras = true	0,361625444
3	password	0,767162647	image	1.104	category	0,28152831	get,Time = true	0,187557312
4	abajo_sistema	0,763315051	bing	1.014	rated_most	0,28152831	Copyright = true	0,181796835
5	aclaran_eterna	0,761963153	configuracion_busqueda	993	recommended_category	0,28152831	[input,submit = [true	0,119986171
6	academico_estudiantes	0,760995878	microsoft	993	jennifer	0,281447213	password = true	0,119986171
7	abono	0,757863489	conexion	928	recommend ed	0,280885202	price = true	0,11451466
8	abrir_tumba	0,731037594	configuracion	873	most	0,28022988	[Security Grade] = S/C	0,104828386
9	accesorios_vehiculos	0,712646082	[Security Grade]	870	rated	0,279600859	Dominio = sp	0,093365377
10	button,submit	0,672346836	privacidad_cookies	816	giving	0,278925195	FAQ = true	0,068154061
11	assassin	0,637993112	reino_unido	789	cumshots	0,278164565	Dominio = om	0,063929039
12	auto_normal	0,629899388	unido	789	funny	0,277721542	Dominio = e/	0,039316537
13	agrado_hectar	0,62625898	italia	778	amateur_anal	0,277047545	SF span peronas|empresas = false	0,258967543
14	agrado_borde	0,615910444	irlanda	750	again	0,276020411	SF sbif = false	0,015163527
15	bing	0,59071557	reino	750	long	0,275744149	Dominio = tv	0,115804985
16	ergo	0,589426044	desactivado	747	latest	0,27566193	Dominio = le	0,101921015
17	grand	0,583130622	preferencias	741	categories_amateur	0,275660203	Dominio = et	0,096919592
18	consejos_seguridad	0,57909791	price	689	couple	0,275660203	cables	0,068735618
19	enhanced	0,577116041	dansk	673	fingering	0,275660203	Dominio = s/	0,065191863
20	anuncios_ayuda	0,558579484	norsk	673	hairy	0,275660203	entretencion	0,051771553

Figura C.1: Análisis de pesos