UNIVERSIDAD DE OVIEDO

ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN

MÁSTER EN INGENIERIA INFORMÁTICA

TRABAJO FIN DE MASTER

SELECCIÓN Y EVALUACIÓN DE ALGORITMOS PARA CLASIFICACIÓN DE DOCUMENTOS

Paula María Vergara García

Mayo 2014

Indicé

Índice de ilustraciones

1.Memoria

1.1.Introducción

La clasificación de documentos consiste en el agrupamiento por temas, conceptos, características o similitud de documentos.

En la actualidad la clasificación de documentos se convierte en un problema por la gran cantidad de información de la que se dispone y la necesidad de acceder a ella. Para ello se dispone de distintos métodos de clasificación, algunos generales y otros más específicos, dependiendo de las características de los documentos a clasificar o de las necesidades del usuario.

El caso que nos ocupa se centra en la clasificación de noticias. No parece un tema muy problemático ya que todos disponemos de medios que nos permiten acceder a las noticias ya clasificadas. Pero si realizamos un pequeño ejercicio y pensamos como se han clasificado las noticias en cualquier publicación, nos encontramos que cada medio de prensa, editor o redactor de la noticia decide la clasificación de la misma y con ello su importancia o posterior tratamiento.

Ahora bien, ampliemos el ángulo de mira y pensemos qué sucede cuando esas noticias se exportan a otros países, donde la clasificación de la que disponemos no resulta útil, a saber: portada, local, nacional, internacional.

Para ello se dispone del estándar IPTC[1], una categorización internacional que ayuda a clasificar las noticias para ser exportadas. En la actualidad este sistema es muy utilizado en los países anglosajones y poco en los de habla hispana, limitando así la difusión de las noticias y expansión del medio de comunicación.

Algunos medios de prensa, como El País[2] en su versión digital, ya utiliza este método de clasificación combinándolos con otros baremos.

En la actualidad existen algunas compañías como Textalytics[3] y Classora[4] que realizan clasificaciones de documentos, incluyendo el estándar IPTC, pero son de pago.

[1] http://www.iptc.org/site/Home/

[2] http://elpais.com/

[3] http://textalytics.com

[4] http://www.classora.com/

Siendo un método de clasificación poco conocido y utilizado, en nuestro idioma, se considera oportuno realizar una investigación sobre su implementación algorítmica y su uso.

1.2.Objetivo

El objetivo de este trabajo es indagar en el campo de la clasificación de documentos en castellano.

Se realizará un análisis de los algoritmos de clasificación y de herramientas que los implementen, con el fin de realizar un análisis y evaluación de dichos algoritmos.

Con los resultados obtenidos se realizara una aplicación donde se implementará el algoritmo de clasificación seleccionado. La aplicación permite seleccionar fuentes de noticias de medios generalistas disponibles a través de Internet (RSS[5]) y realizar una clasificación en distintas categorías según los contenidos de las noticias.

De esta forma el trabajo se ha dividido en dos partes, una de estudio científico - técnico y otra en el diseño de un servicio.

[5] http://www.rss.nom.es/

2.Estudio científico – técnico

2.1.Introducción

El presente proyecto pretende obtener un algoritmo de clasificación de documentos lo más eficiente posible para su uso en la clasificación de noticias RSS a través del análisis de diferentes algoritmos de clasificación.

Para ello se realizará:

- Análisis de distintos algoritmos de clasificación basados en categorías predeterminadas.
- Selección de algoritmo de clasificación en base al análisis de las características de caso a tratar y de las herramientas disponibles para realizar el trabajo.
- Elaboración de conjuntos de entrenamiento para posteriormente poder analizar la bondad del algoritmo seleccionado.
- Evaluación y análisis de los distintos modelos. Para la elección del mejor modelo para la posterior clasificación de noticias.

2.2.Estado del arte

2.2.1.Introducción

Para alcanzar el objetivo deseado de clasificación de documentos, debemos partir del análisis de las alternativas que nos ofrece el estado del arte en los siguientes campos:

1. Elección del tipo de aprendizaje.
2. Elección de la categorización a utilizar.
3. Elección del algoritmo de clasificación según el esquema de aprendizaje.
4. Análisis de herramientas que faciliten:
    a. Transformación de los documentos que componen los datos a clasificar en un formato comprensible para el sistema clasificador.
    b. Entrenamiento de los datos en base a las categorías predefinidas para la creación de modelos.
    c. Categorización de los datos por clasificar con ayuda del modelo creado previamente.
    d. Evaluación de los resultados obtenidos en la fase de test.

En la Ilustración 1, se puede ver un esquema que indica los pasos que se seguirán el en proceso de análisis de las distintas alternativas de las que se dispone:

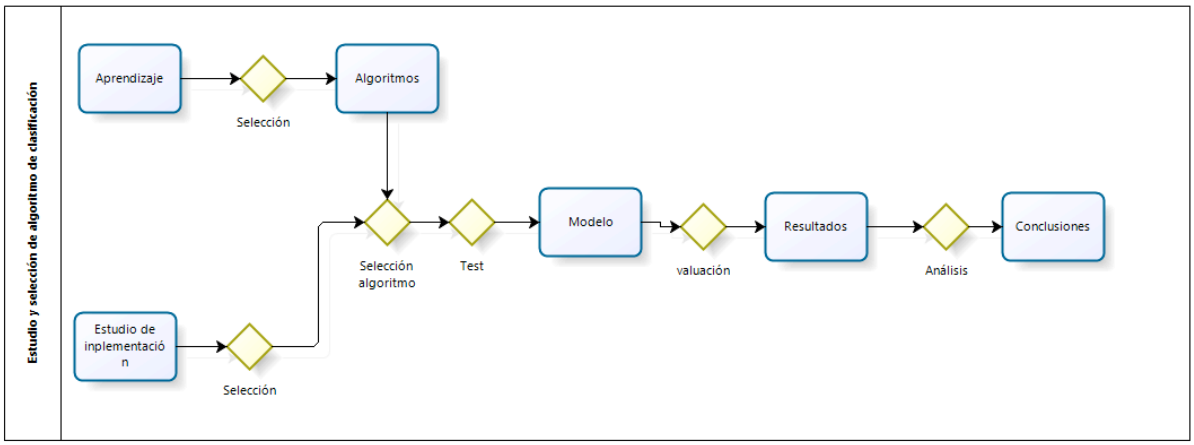


Ilustración 1- Esquema a seguir en el estudio y selección de algoritmos de clasificación

- Inicialmente se realizara un estudio de los distintos métodos de aprendizaje. Se concluirá cuál de ellos se adapta mejor a las necesidades del trabajo.
- Se realizara una presentación de los distintos algoritmos de clasificación que se ajusten al aprendizaje seleccionado.
- Se realizara un estudio de los medios que se dispone para la realización de la implementación.
- Con todo ello se decidirá que algoritmo de clasificación se utilizará para realizar la clasificación.
- Se seguirán los pasos definidos por la clasificación y el algoritmos seleccionados para obtener los datos a evaluar.
- Se realizará el análisis y evaluación de los datos obtenidos en la clasificación.

2.2.2.Aprendizaje automático (Máquina de aprendizaje)

En este apartado se tratará el tema del aprendizaje y se dividirá en tres partes:

- Introducción: donde se darán algunas definiciones sobre el aprendizaje.
- Se realizará una presentación de los distintos tipos de aprendizaje.
- Y para terminar se tomará una decisión sobre el tipo de aprendizaje a utilizar en el trabajo.

2.2.2.1.Introducción

También llamado aprendizaje automático, es una rama de la inteligencia artificial cuyo objetivo es desarrollar técnicas que permitan a las computadoras aprender.

Algunas de las aplicaciones de las máquinas de aprendizaje son: motores de búsqueda, diagnósticos médicos, detección de fraude en el uso de tarjetas de crédito, clasificación de secuencias de ADN, reconocimiento del habla y del lenguaje escrito, robótica y un largo etc.

En el proceso del aprendizaje automático no hay que olvidar la importancia de la intuición humana, y general una interacción hombre- máquina.

Se utilizan distintos algoritmos en el aprendizaje automático para generar conocimiento y mejorar el rendimiento de los sistemas de computación. Estos algoritmos se clasifican según el tipo de aprendizaje, supervisado, semi-supervisado y no supervisado.

2.2.2.2.Tipos de aprendizaje automático

El aprendizaje automático es la base para la construcción de clasificadores automáticos. Este aprendizaje se puede ser de tres tipos:

- **Aprendizaje supervisado**: a partir de una colección de entrenamiento se aprenden las características que debe cumplir un documento para pertenecer a una u otra clase, creando posteriormente el clasificador o modelo. Una vez terminada esta fase de entrenamiento, el clasificador final está definido, el cual se utilizara para la categorización de documentos de los que no se conoce su clase.
- **Aprendizaje semi-supervisado**: la fase de creación del clasificador utiliza la colección de entrenamiento como base, pero se sigue refinando con documentos sin clasificar. En este caso el número de los documentos sin clasificar suele ser mucho mayor que el número de los documentos ya clasificados. Este tipo de aprendizaje puede ayudar en el caso de tener un número pequeño de documentos preclasificados, pero por lo general es más crítica la creación de un buen clasificador.
- **Aprendizaje no supervisado**: En este caso no se dispone de documentos clasificados. La clasificación es una agrupación de documentos en cluster.

2.2.2.3.Decisión

En el caso a tratar se dispone de un conjunto de entrenamiento, por lo que se podría realizar un aprendizaje supervisado o semi-supervisado, en un principio se realizará un aprendizaje supervisado. Dependiendo de los resultados de evaluación se podrá considerar realizar una realimentación en el conjunto de entrenamiento con los documentos clasificados realizando así un aprendizaje semi-supervisado.

La clasificación supervisada se podría definir como la asignación de una o varias categorías predefinidas sobre un grupo de documentos (instancias) a clasificar. En general se puede dividir en las siguientes fases:

- **Representación**: Los documentos que componen los datos a clasificar deben ser transformados a un formato compresible para el sistema de clasificación a utilizar.
- **Clasificación**:
    - **Entrenamiento**: permite obtener la descripción de las categorías con ayuda de la colección de documentos previamente clasificados. Lo que se denomina "modelo".
    - **Test:** una vez obtenido el modelo, se podrá predecir las categorías de los documentos por clasificar.
- **Evaluación**: Una vez realizada la clasificación se realizara un análisis de las clasificaciones obtenidas, para poder evaluar la calidad de la clasificación.

La Ilustración 2 muestra gráficamente las fases de la clasificación supervisada.

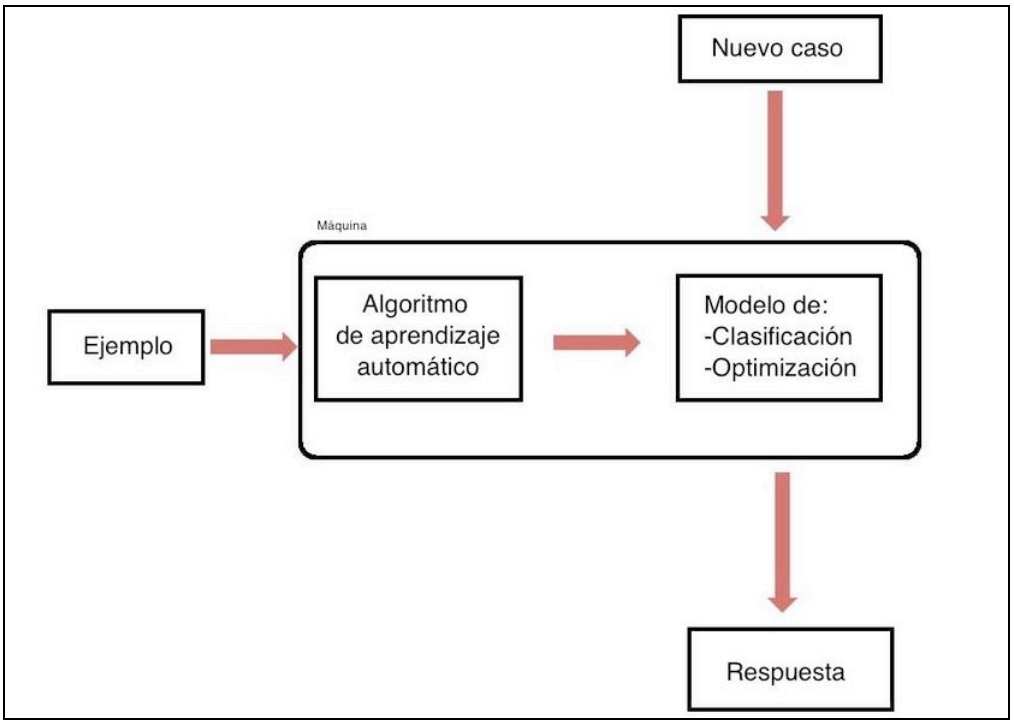


Ilustración 2- Proceso de clasificación supervisada

2.2.3.Categorización

Una vez seleccionado en el apartado anterior, en sistema de aprendizaje supervisado el cual necesita para su implementación un conjunto de ejemplos previamente etiquetados por categorías:

- Presentación de distintas opciones de categorización.
- Toma de decisión sobre la categorización que se utilizará en el trabajo.
- Presentación de la opción seleccionada.

2.2.3.1.Introducción

En esta sección se realiza una introducción a la definición de categorización y se presentarán algunas de las opciones de categorización que se utilizan hoy en día.

La categorización consiste en el etiquetado de documentos según unas categorías dadas. Según el tipo de documentos a clasificar, existen distintos métodos:

- Para archivos y clasificación de documentos existen métodos estándar (norma AENOR) : alfabético, cronológico, decimal, geográfica por materias, etc.
- Usenet[6]: Grupos de noticias se organizan las noticias en 8 categorías base:
    - comp.* Temas de Computadoras
    - humanities.* Temas de humanidades
    - misc.* Temas misceláneos

[6] http://es.usenet.nl/

- news.* Temas relacionados con los newsgroups mismos. Esta jerarquía no es para hacer cobertura noticiosa, sino para discusiones relacionadas con la administración de Usenet en particular.
- rec.* Recreación y entretenimiento
- sci.* Temas científicos
- soc.* Temas culturales y sociales

- Tweet Category[7]: Una aplicación para clasificar y ordenar los tweets por temas que previamente a seleccionado el usuario.
- Los periódicos digitales tienen sus propios criterios, un ejemplo, "noticias.com[8]":
  - Portada
  - Internacional
  - España
  - Política
  - Deportes
  - Economía
  - Ciencia
  - Cultura
  - Sociedad
  - Tecnología
- Estándar IPTC: asociación que agrupa a varias de las empresas de comunicación más importantes del mundo, incluyendo a agencias de noticias, medios digitales y prensa escrita. Clasifica las noticias en 18 categorías básicas.

2.2.3.2.Decisión

Como se ha visto en el apartado anterior, no existe un método general o estándar de categorización que se pueda aplicar a las noticias en castellano, cada medio de difusión utiliza el suyo propio.

Pero estos criterios de categorización no son útiles a al hora de exportar las noticias, para ello se necesita un criterio global, un método utilizado en todos los medios de prensa y en todos los países, de esta manera se realiza un acceso rápido y ordenado a la información.

Por ello para este trabajo se ha optado por una categorización definida por el estándar IPTC, muy utilizado en el mundo anglosajón, pero poco utilizado en castellano, pero con una gran expansión de futura ya que resulta muy útil a la hora de la comunicación digital.

2.2.3.3.IPTC

Una ver seleccionado el método de categorización se realizará una presentación del mismo:

El International Press Telecommunication Council (IPTC), con sede en Reino Unido, es una asociación que agrupa a varias de las empresas de comunicación más importantes del mundo,

[7] https://twitter.com/tweetcategory

[8] http://www.noticias.com/noticias/clasificacion

incluyendo a agencias de noticias, medios digitales y prensa escrita. Su objetivo inicial era salvaguardar los intereses en las telecomunicaciones de la prensa mundial, desarrollan y mantienen estándares técnicos para mejorar el intercambio de noticias que son usadas por las mayores agencias de noticias del mundo.

Este organismo ofrece un lenguaje internacional, homologado para documentar la información, imprescindible para compartir recursos entre los medios de comunicación y responder ágilmente a la creciente demanda de productos temáticos y personalizados que estas empresas sirven a sus clientes.

IPTC genera un modelo estándar para el intercambio de información de todo tipo (noticias, textos, imágenes, vídeos, eventos, etc.). Este estándar se denomina Information Interchange Model (IIM). La aceptación del IIM fue todo un éxito en el mundo de la fotografía (tanto profesional como amateur), si bien en el mundo de la prensa digital apenas tuvo repercusión, utilizándose al principio únicamente en agencias de noticias de algunos países europeos.

No obstante, poco a poco la evolución de la Web Semántica y la mayor acogida general de los estándares de clasificación y representación de la información están haciendo que también los estándares IIM de IPTC vuelvan a estar entre las prioridades de varias compañías del sector en todo el mundo.

Una de las principales ventajas a la hora de explotar las recomendaciones del IPTC reside en poder vender noticias a otros medios de comunicación anotándolas con categorías y etiquetas estándares. Dentro del IIM, el IPTC generó un listado pormenorizado y flexible de categorías para clasificar textos y noticias de todo tipo. Este listado, está formado por una clasificación jerárquica con 1400 categorías, está organizada en tres niveles y tiene 18 categorías principales (ilustración 3). Cada categoría incluye un código numérico único (code) y una descripción que forma parte de la etiqueta (label). Tiene un valor estratégico incalculable para intercambiar información y así abrir nuevas vías de negocio.

Código	Etiqueta	Descripción
01000000	arte, cultura y espectáculos	Asuntos pertinentes al avance y refinamiento de la mente humana, intereses, habilidades, gustos y emociones.
02000000	policía y justicia	Establecimiento y/o tratados de las reglas de comportamiento de la sociedad, el fortalecimiento de estas reglas, quiebres de las reglas y penas de los ofensores.
03000000	catástrofes y accidentes	Accidentes humanos y eventos de la naturaleza resultantes en pérdida de vidas o daños a criaturas vivientes.
04000000	economía, negocios y finanzas	Todos los asuntos concernientes a la planificación, producción e intercambio de riqueza.

05000000	educación	Todas las formas de lograr conocimiento desde el nacimiento a la muerte de las personas.
06000000	medio ambiente	Todos los aspectos de protección, daño y condiciones del ecosistema del planeta Tierra y sus alrededores.
07000000	salud	Todos los aspectos pertinentes al bienestar psicológico y mental de los seres humanos.
08000000	interés humano	Cosas acerca de individuos, grupos, animales u objetos.
09000000	mano de obra	Aspectos sociales, organizaciones, reglas y condiciones que afectan al empleo y la generación de riqueza o provisión de servicios. Soporte económico a los desempleados.
10000000	estilo de vida y tiempo libre	Competencias que generalmente entretienen.
11000000	política	Ejercicio o lucha local, regional, nacional o internacional por el poder, y relaciones entre entidades rectoras y los estados.
1200000	religión y credos	Todos los aspectos de la existencia humana que involucran teología, filosofía, ética y espiritualidad.
13000000	ciencia y tecnología	Todos los aspectos relacionados con la interpretación humana de la naturaleza y el mundo físico, y el desarrollo y la aplicación de este conocimiento.
14000000	asuntos sociales	Aspectos del comportamiento humano que afectan a la calidad de vida.
15000000	deporte	Ejercicio competitivo, que involucra esfuerzo físico. Organizaciones y cuerpos involucrados en dichas actividades.
16000000	disturbios, conflictos y guerra	Actos de protesta y/o violencia motivados por conflictos sociales o políticos.
17000000	meteorología	El estudio, informe y predicción de fenómenos meteorológicos.
22000000	multitemático	Resúmenes breves, previsiones, efemérides, mensajes

Ilustración 3- Primer nivel de categorización del estándar IPTC

Dentro del primer nivel de categorización se encuentran otros dos niveles, a continuación se muestran algunos ejemplos, en concreto los relacionados con la informática. (Ilustración 4)

1º nivel	2º nivel	3º nivel
Arte, cultura y espectáculos	Internet	
Justicia e interior	Criminalidad/sucesos	Delitos informáticos
Economía, negocios y finanzas	Informática y tecnologías	• Hardware • Interconexión • Tecnología satelital • Semiconductores • Software • Equipos de telecomunicaciones • Telecomunicaciones • Seguridad • Tecnología inalámbrica
	Comercio	Comercio electrónico
	Medios de difusión	Publicación en internet

Ilustración 4- Ejemplo de las posibles categorizaciones relacionadas con la informática

En el anexo A se encuentran todas las categorías en castellano del estándar IPTC.

2.2.4.Algoritmos de clasificación automática supervisada

Como introducción a las posibles herramientas de implementaciones se presentan distintos algoritmos de clasificación supervisada que podrían implementar nuestra clasificación.

- **Clasificación probabilística**: basada en el teorema de Bayes. , también conocido como teorema de la probabilidad condicionada.

  Sea $\{A_1, A_2,\ldots,A_i,\ldots,A_n\}$ un conjunto de sucesos mutuamente excluyentes y exhaustivos, y tales que la probabilidad de cada uno de ellos es distinta de cero (0). Sea B un suceso cualquiera del que se conocen las probabilidades condicionales $P(B/A_i)$. Entonces, la probabilidad $P(A_i/B)$ viene dada por la expresión:

$$P(A_i/B) = \frac{P(B/A_i)P(A_i)}{P(B)}$$

  donde:

  - $P(A_i)$ son las probabilidades a priori.
  - $P(B/A_i)$ es la probabilidad de B en la hipótesis A_i.
  - $P(A_i/B)$ son las probabilidades a posteriori.

- **Arboles de decisión:** Son árboles cuyos nodos están etiquetados por términos, las ramas salientes están etiquetadas por los pesos de estas y las hojas corresponden a las categorías. De esta forma se recorre el árbol de arriba abajo para cada uno de los documentos, hasta llegar a un ahoja y asignar una categoría. Esta técnica de clasificación apenas es utilizada en la actualidad.

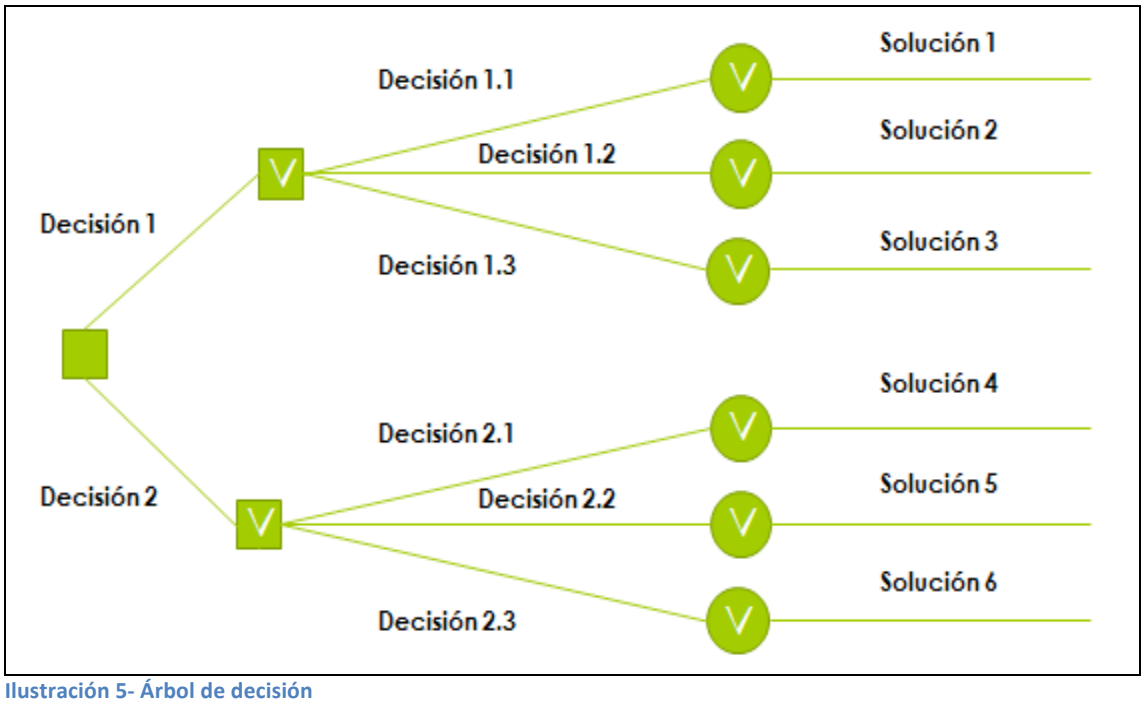


Ilustración 5- Árbol de decisión

- **Clasificación basada en reglas:** Similar a clasificación con árboles de decisión, pero con la definición de una serie de reglas para cada una de las categorías. De esta forma se consigue una forma más compacta de clasificación de los documentos sin recorrer el árbol completo. Necesita la implicación de un experto para la creación de las reglas.
- **Método de regresión:** Predicen una o más variables continuas, como las pérdidas o los beneficios, basándose en otros atributos del conjunto de datos. Estos métodos se basan en la creación de una aproximación a la función ideal de clasificación con valores reales en vez de binarios, mediante la que se ajusta a los valores de entrenamiento. En este método se asignan dos vectores a cada documento, uno de entrada que representa los pesos de los términos y otro de salida que representa los pesos de las categorías. Con estos vectores se crean unas matrices para que posteriormente sean procesadas. La clasificación se realiza mediante la multiplicación de dichas matrices, produciendo un gran coste computacional, Aunque se obtengan buenos resultados.
- **Clasificación lineal:** Este método utiliza un vector para cada categoría existente y otro para cada documento, los cuales está formado por pesos de cada uno de los términos sobre cada categoría y documento. Entre los métodos lineales se puede distinguir los que ejecutan por lotes, donde se analiza toda la colección de entrenamiento a la vez, y los métodos incrementales, donde se empieza a construir el clasificador con el primer documento de entrenamiento y se va refinando conforme se van analizando nuevos documentos.
- **Redes neuronales:** En el caso del aprendizaje supervisado se proporciona a la red los ejemplos como entrada y como salida la clasificación deseada para los ejemplos, el algoritmo ajustará los pesos y cambiara los parámetros de la red para minimizar el error y proporcionar la salida deseada.

- **Clasificación basada en ejemplos:** El método más utilizado es el algoritmo k-NN (k-Nearest Neighborhood) que a la hora de decidir qué documento pertenece a cierta categoría, selecciona los k documentos más similares al utilizado como ejemplo. Este método requiere de la definición inicial de un valor de k.
- **Máquinas de vector soporte (SVM):** estructuras de aprendizaje basadas en la teoría estadística del aprendizaje. Se basan en transformar el espacio de entrada en otro de dimensión superior (infinita) en el que el problema puede ser resuelto mediante un hiperplano óptimo (de máximo margen).

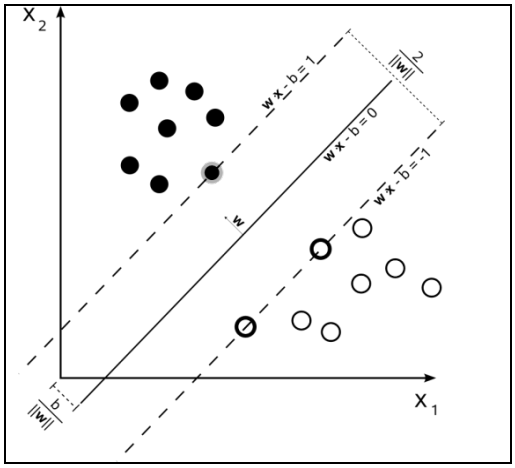


Ilustración 6- Máquina vector soporte

- **Otros:**
  - Algoritmos genéticos.
  - Redes de inferencia bayesiana.
  - Modelado de entropía máxima.
  - ….

2.2.5.Herramientas de implementación

2.2.5.1.Introducción

Una vez seleccionado el tipo de clasificación que se va a realizar (algoritmo de clasificación supervisada), se necesitaran herramientas que permitan implementar las fases del proceso de clasificación supervisada:

- **Representación**: Los documentos que componen los datos a clasificar deben ser transformados a un formato compresible para el sistema de clasificación a utilizar.
- **Clasificación**:
  - **Entrenamiento**: permite obtener la descripción de las categorías con ayuda de la colección de documentos previamente clasificados. Lo que se denomina "modelo".
  - **Test:** una vez obtenido el modelo, se podrá predecir las categorías de los documentos por clasificar.

- **Evaluación**: Una vez realizada la clasificación se realizara un análisis de las clasificaciones obtenidas, para poder evaluar la calidad de la clasificación.

Para realizar este trabajo se han utilizado distintas herramientas de uso gratuito y código libre.

Por una parte se necesitará realizar tratamientos sobre documentos (como extracción de datos o parsing, normalización léxica y semántica, etc), para su posterior análisis, clasificación y evaluación. Para realizar estas operaciones se encuentran disponibles en el mercado distintas herramientas como procesadores de lenguaje natural o normalización (basados en el algoritmo de stemming de Porter[9]), implementación de algoritmos de clasificación: Websom (clustering), evaluación del modelo (weka, Knime, MatLab, Tanagra, R, …).

Hay que tener en cuenta que muchas de ellas no son de licencia libre, otras utilizan lenguajes propios, en el caso de los evaluadores se necesita parsear los ejemplo de forma específica, no resultando esta tarea muy sencilla.

2.2.5.2.Decisión

Se ha optado por Apache Software Foundation (ASF)[10]. Ya que proporciona todas las herramientas necesarias para realizar el trabajo y se complementan entre sí, compartiendo el mismo lenguaje y soportando librerías compatibles, proporcionando una solución integrada.

Otro motivo fue que las librerías Apache son muy usadas por la comunidad investigadora en el campo de la semántica, se dispone de medios como amplia documentación y foros de discusión, entre otros.

2.2.5.3.Apache Software Foundation (ASF): Hadoop, Mahout, Lucene, Tika

Tomada la decisión de utilizar las herramientas proporcionadas por ASF, se procederá a realizar una introducción de las herramientas que se utilizarán en el trabajo, analizando sus características y propiedades mas notables.

Las herramientas utilizadas serán:

- Hadoop
- Lucene
- Tika
- Mahout

9 http://tartarus.org/~martin/PorterStemmer/
10 https://www.apache.org/foundation/

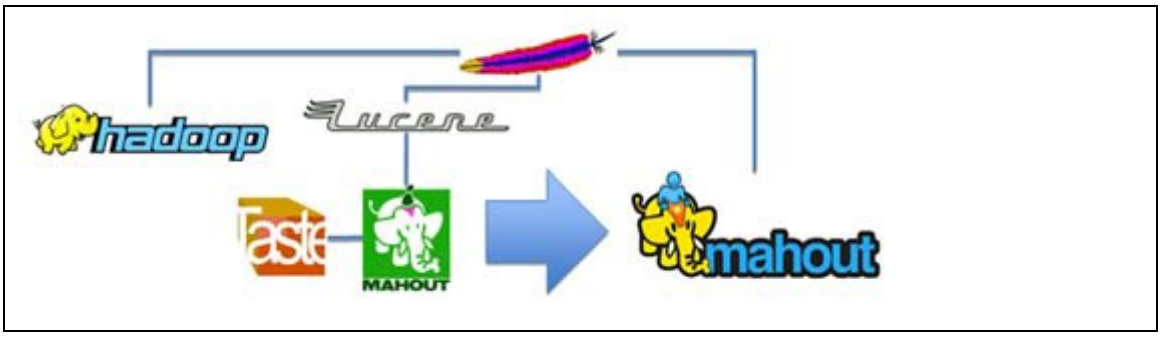


Ilustración 7-Proyectos de ASF utilizados en el trabajo

Apache Software Foundation (ASF) es una organización sin ánimo de lucro (en concreto, una fundación) creada para dar soporte a los proyectos de software bajo la denominación Apache, entre los que se incluye el popular servidor HTTP Apache. La ASF se formó a partir del llamado Grupo Apache. Esta fundación abarca gran cantidad de proyectos desarrollados por desarrolladores voluntarios. ASF proporciona todo un conjunto de aplicaciones y desarrollos compatibles entre sí, para facilitar el análisis y clasificación de documentos. Dotando de una solución integral al problema propuesto.

A continuación comentaremos los proyectos proporcionados por ASF que se utilizarán para el desarrollo del trabajo.

2.2.5.4.Hadoop

Es una infraestructura digital de desarrollo creada en código abierto bajo licencia Apache, un proyecto construido y utilizado por una gran variedad de programadores utilizando Java. Permite desarrollar tareas muy intensivas de computación masiva, dividiéndolas en pequeñas piezas y distribuyéndolas en un conjunto todo lo grande que se quiera de máquinas: análisis de petabytes de datos en entornos distribuidos formados por muchas máquinas sencillas. Esta basado en los documentos Google para MapReduce y Google File System (GFS).

2.2.5.4.1.MapReduce

Modelo de programación funcional en paralelo diseñado para escalabilidad y tolerancia a fallos en grandes sistemas de commodity hardware:

• Basado en la combinación de operaciones Map y Reduce

• Diseñado originalmente por Google

• Usado en múltiples operaciones

• Manejo de varios petabytes diarios

• Popularizado por la implementación open source Apache Hadoop

• Usado por Facebook, Last.fm, Rackspace, yahoo, Amazon Web Services…

Ejemplos de algunas aplicaciones de MapReduce:

- En Google:
    - Construcción de índices para el buscador (pagerank)
    - Clustering de artículos en Google News
    - Búsqueda de rutas en Google Maps
    - Traducción estadística
- En Facebook:
    - Minería de datos
    - Optimización de ads
    - Detección de spam
    - Gestión de logs
- En I+D+i:
    - Análisis astronómico
    - bioinformática
    - física de partículas
    - simulación climática
    - procesamiento del lenguaje natural

Organizaciones que usan Hadoop:

- A9.com
- AOL
- Booz Allen Hamilton
- EHarmony
- eBay
- Facebook
- Fox Interactive Media
- Freebase
- IBM
- Análisis y benchmark de Hadoop
- Análisis y benchmark de Hadoop 5 de 11
- ImageShack
- ISI
- Joost
- Last.fm
- LinkedIn
- Meebo
- Metaweb
- Mitula15
- The New York Times
- Ning
- Powerset (ahora parte de Microsoft)
- Rackspace
- StumbleUpon16

- Tuenti
- Twitter
- Veoh
- Zoosk
- 1&1

2.2.5.5.Lucene

Apache Lucene es una herramienta para la recuperación de información (IR), existen versiones para distintos lenguajes de programación: Java, Delphi, Perl, C#, C++, Python, Ruby y PHP. Es independiente del formato del fichero, puede indexar textos que se encuentren en PDF, páginas HTML, documentos de Microsoft Word. Lucene ha sido ampliamente usado por su utilidad en la implementación de motores de búsqueda (Nutch[11], Akamai[12]).

Principales funcionalidades:

- Proporciona la capacidad de buscar en muchos idiomas.
- Indexa cualquier archivo de texto, tal como HTML, o cualquier archivo que se pueda convertir en texto.
- Clasifica la búsqueda de manera que se muestren los mejores resultados primero
- Realiza búsqueda booleana y por frase.
- Permite búsqueda por campo (ej., las búsquedas se pueden enviar por títulos, autores, contenidos, etc.).
- Permite búsquedas en un gran rango de fechas para que los usuarios puedan tener acceso a información sujeta a plazos determinados.

[11] http://nutch.apache.org/

12 http://spanish.akamai.com/enes/

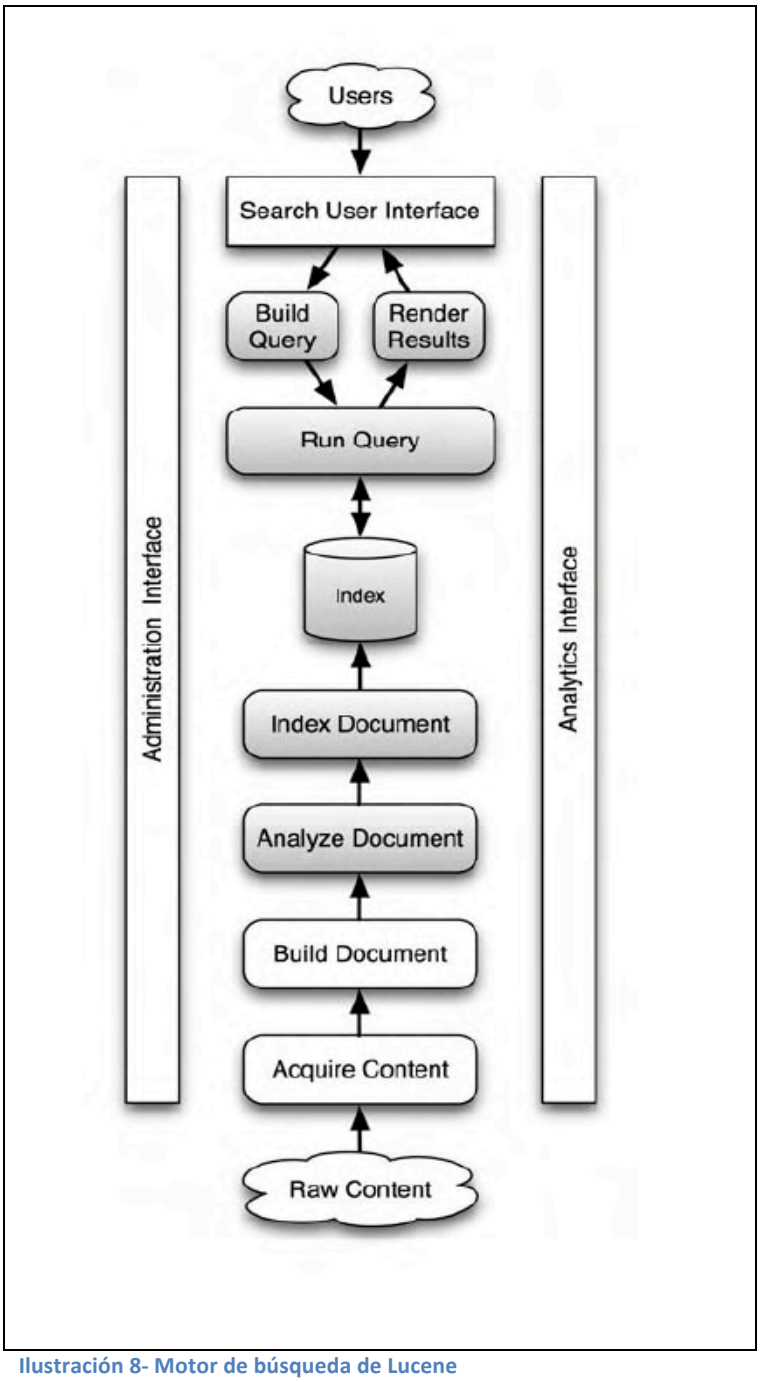


Ilustración 8- Motor de búsqueda de Lucene

El motor de búsqueda es similar para la gran variedad de aplicaciones a las que Lucene presta servicio, búsqueda de contenidos en archivos locales, búsqueda de catálogos de productos en servicios web para varios usuarios, búsqueda de archivos o documentos en intranet de empresas, etc.

Lucene se encarga principalmente del indexado y búsqueda (parte oscura del gráfico del motor de búsqueda), dejando las demás partes a cargo del usuario.

De abajo arriba del motor de búsqueda, empezamos por la indexación, esto es, el proceso de análisis de los documentos para la obtención de una representación de los mismos.

2.2.5.5.1.Indexación

El indexado es el proceso de recuperación de datos contenidos en un documento, en este proceso se crea un índice que guarda la posición de los datos obtenidos.

Atendiendo a la gráfica el indexado se divide en :

- Acquire content: Obtención a los contenidos de búsqueda. Dependiendo de sistema al que accedamos puede ser una tarea trivial (documentos XML sistema de archivos o base de datos bien organizada) o un quebradero de cabeza si la información está dispersa.
- Build document: Divide en pequeños contenidos llamados normalmente documentos. Antes tendrá que ejecutar filtros para extraer las partes no importantes. Este proceso no es proporcionado por Lucene, pero si por Apache Tika que proporciona un filtrado de documentos.
- Analyze document: Divide los textos en tokens, elementos individuales o lo que es lo mismo, palabras. También se encarga del control de plurales o formas verbales (algoritmo de stemming de Porter), Lucene dispone de gran variedad de analizadores que permiten el control de este proceso.
- Index document: Añade el documento para el índice.

2.2.5.5.2.Búsqueda

La búsqueda es el proceso para buscar palabras en un índice para encontrar documentos en las contengan.

- Search user interface: La interface de usuario es lo que el usuario realmente ve cuando solicita la búsqueda.
- Build query: Generar la consulta es traducir la solicitud en objetos de consulta del motor de búsqueda.
- Search query: la consulta de búsqueda consulta el índice de búsqueda y se recuperan los documentos que coincidan con la consulta.
- Render resultados: Muestra de los resultados al usuario.

2.2.5.6.Tika

Apache Tika es un subproyecto de Apache Lucene. Su objetivo es detectar, extraer y analizar los contenidos de distintos tipos de documentos como HTML, XML, RTF, PDF, etc, mediante librerías de parseo. Para realizar esta tarea dispone de una biblioteca de APIs de extracción de datos, la cual esta formada por analizadores que permiten extraer el texto requerido. Estos analizadores no son creados por Tika, sino que se basa en proyectos de código abierto ya existentes, esta biblioteca va creciendo a medida que aparecen más analizadores que amplían la biblioteca. Tampoco es necesario indicarle el tipo de documento o analizador necesario para la extracción de los datos, si el tipo de documento es conocido, encontrara y aplicara el analizador correspondiente de su biblioteca de APIs. A continuación se muestran algunos ejemplos de la biblioteca de APIs de la que dispone Tika:

Formato	Biblioteca
Microsoft Office OLE2 Compound Document Format (Excel, Word, PowerPoint, Visio, Outlook)	Apache POI
Microsoft Office 2007 OOXML	Apache POI
Adobe Portable Document Format (PDF)	PDFBox
Rich Text Format (RTF) – currently body text only (no metadata)	Java Swing API (RTFEditorKit)
Plain Text	ICU4J library
HTML	CyberNeko library
XML	Java's javax.xml clases
ZIP Archives	Java's builtin ZIP clases
TAR Archives	Apache Ant
GZIP compression	Java's built-in support (GZIPInputStream)
BZIP2 compression	Apache Ant
Image formats (metadata only)	Java's javax.imageio clases
Java class files	ASM library (JCR-1522)
….	…..

Ilustración 9- Ejemplos de tipos de formatos que acepta Tika y que bibliotecas utiliza para su parseo

2.2.5.7.Mahout

Apache Mahout es una máquina de aprendizaje (machine learning) cuyo objetivo consiste en construir bibliotecas escalables de aprendizaje automático. Construido sobre el potente paradigma map/reduce del proyecto Apache Hadoop, Mahout permite resolver problemas como clustering, filtrado colaborativo y clasificación de terabytes de datos sobre miles de ordenadores.

En este trabajo se utilizará las librerías que Mahout proporciona para la clasificación. Realizando con su ayuda las fases de entrenamiento, test y evaluación.

El proceso de clasificación comienza con la creación del modelo. En el siguiente grafico (Ilustración 10) se pueden ver los pasos que realiza Mahout para obtener el modelo y poder realizar la clasificación:

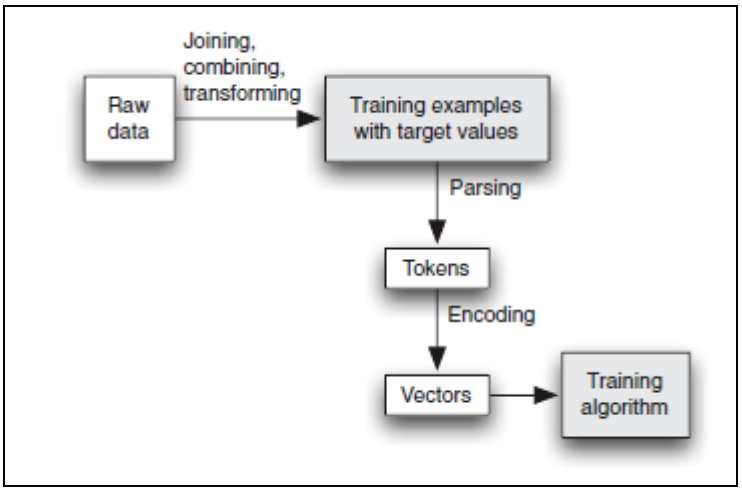


Ilustración 10- Estructura de Mahout

Los algoritmos de clasificación requieren los datos de entrada en forma de vector, por ello los datos de entrada se deben almacenar en un único registro para el posterior análisis y vectorización.

Raw data son los datos de entrada se ordenan en un registro con campos idénticos, estos campos pueden ser de cuatro tipos, continuos, categóricos, palabras simples o texto simple. Para su posterior análisis (parsing), el análisis se puede realizar con distintas herramientas, en nuestro caso utilizaremos Lucene. La codificación será realizada por Mahout. (Mahout proporciona el "encode" y Lucene el "parser").

Los vectores de codificación pueden ser de tres tipos, dependiendo del algoritmo que se vaya a utilizar.

Planteamiento (approach)	Beneficios	Costes	Usos
Utiliza un vector celda (one vector cell) por palabra, categoría o variable continua.	No implica colisiones y se puede revertir fácilmente	Requiere dos pasos, uno de asignación de celdas y otro para establecer valores, los vectores pueden tener diferentes longitudes.	clustering
Representa un vector como un conjunto de palabras	No implica colisiones se realiza en un paso.	Difícil de usar el álgebra lineal, dificultad a la hora de representar valores continuos y debe formatear los datos en un formato especial no-vectorial	Naive Bayes
Función hashing	El tamaño del vector se fija de antemano, es una buena opción para las primitivas de álgebra lineal. Se realiza en un paso.	Colisiones, interpretación del modelo.	Algoritmos de regresión lineal, SGD

Ilustración 11- Mahout: Tipos de vectores de codificación , características y usos

El siguiente paso sería analizar los distintos algoritmos de clasificación que proporciona Mahout.

Dentro del amplio número de algoritmos de los que se dispone, se realizará una presentación de los algoritmos de clasificación automática que utilizan aprendizaje supervisado y que son utilizados por Mahout:

Tamaño del modelo	Algoritmo Mahout	Modelo de ejecución	características
De pequeño a medio	Stochastic gradient descent (SGD)	Secuencial, lineal, incremental	Usa todo tipo de variables predictivas, eficiente con modelos de entrenamiento adecuados.

De medio a grande	Support vector machine (SVM)	Secuencial	Experimental
	Naive Bayes	Paralelo	Eficiente cuando el conjunto de datos es demasiado grande para SGD o SVM
	Complementary naive Bayes	Paralelo	Eficiente con conjuntos de datos demasiado grandes para SGD, pero con limitaciones.
De pequeño a medio	Random forest	Paralelo	Su utilización no es todavía muy amplia.

Ilustración 12- Caracteristicas de los algoritmos de clasificación implementados por Mahout

La principal diferencia entre los algoritmos es el coste del entrenamiento y el tamaño del conjunto de entrenamiento, lo que definirá la eficiencia a la hora de realizar los análisis.

En Mahout, SGD y Random forest hacen un buen uso de las variables continuas. Otros algoritmos no pueden utilizar las variables continuas, como Naive Bayes y Naive Bayes complementario.

El comportamiento del algoritmo SVM es similar al SGD ya que los dos son secuenciales, pero el algoritmo SGD será más lento cuando se tengan grandes conjuntos de datos.

Mahout recomienda que si te tienen un conjunto de ejemplos mayor de 100.000 y una sola categoria, se utilice Bayes, en otro caso SGD.

2.2.5.7.1.Medidas de evaluación

A la hora de la evaluación se dispone de gran número de variables de medición. Para el caso que nos ocupa Mahout proporciona las siguientes métricas:

Metric	Supported by class
Percent correct	CrossFoldLearner
Confusion matrix	ConfusionMatrix, Auc
Entropy matrix	Auc
AUC	Auc, OnlineAuc, CrossFoldLearner, AdaptiveLogisticRegression
Log Likelihood	CrossFoldLearner

Ilustración 13- Medidas de evaluación

- AUC: Área bajo la curva ROC. Sin meterse en definiciones matemáticas e intentando dar una definición sencilla, podríamos definir la curva ROC como una gráfica resumen de toda la información de clasificación. A partir de esta gráfica se puede

definir un área (AUC). AUC es la porción de un área de lado 1 y su valor esta comprendido entre 0 y 1.Es útil en la evaluación de modelos que producen un resultado continuo con una variable binaria.

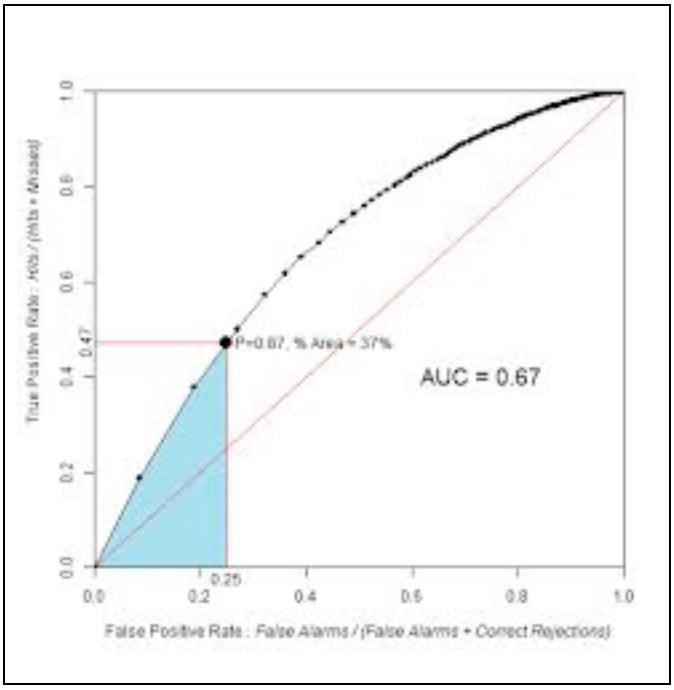


Ilustración 14- Representación gráfica de AUC

En la gráfica se puede ver una representación de la curva ROC y su relación con AUC.

- Matriz de confusión: ordena todos los casos del modelo en categorías, determinando si el valor de predicción coincide con el valor real.
- Precisión (AverageCorrect): La precisión es lo cerca que los valores medidos están unos de otros. Porcentaje de instancias bien clasificadas. Precisión se refiere a la dispersión del conjunto de valores obtenidos de mediciones repetidas de una magnitud. Cuanto de lo que se da por positivo es realmente positivo. Sus valores están entre 0 y 1, la precisión será mejor cuanto más cerca este de 1. Un buen clasificador deberá superar el 0.5.

  Exactitud (Accuracy): La exactitud se refiere a cuán cerca del valor real se encuentra el valor medido.

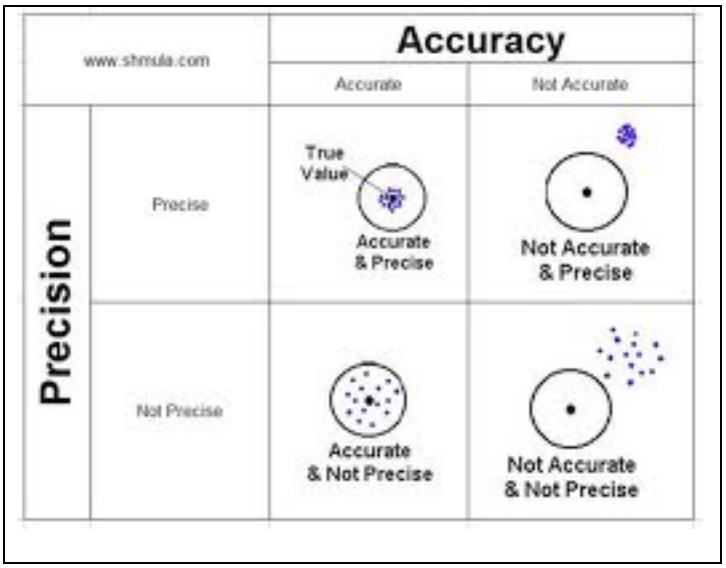


Ilustración 15- Relación entre la precisión y exactitud

- Log-likelihood (averagLL): Representa el valor de la función de verosimilitud en los parámetros, útil para la interpretación del ratio de verosimilitud. Es la transformación logarítmica de la función de verosimilitud evaluada en los estimadores de máxima verosimilitud. Mide en qué grado el modelo se ajusta a los datos, cuanto menor sea su valor, mejor es el ajuste. Sus valores son 0 o menor de 0.
- Máximo log-likelihood, de todos los modelos nos quedamos con el que nos da mayor verosimilitud. Se llama criterio de máxima verosimilitud.

  Loglikelihood tiene un valor máximo de 0 y no hay límite en de cuánto lo negativo que puede llegar a ser. Para clasificadores de alta precisión, el valor del promedio del logaritmo debería ser cercano al porcentaje medio correcto loglikelihood promedio debe estar cerca de la ciento promedio correcto para el clasificador, multiplicado por el número de categorías objetivo. Para clasificadores menos precisos, especialmente aquellos con muchas categorías objetivo, el porcentaje promedio correcto tiende a ir a 0, por lo que es difícil comparar los clasificadores.

2.2.6.Conclusiones

Se seguirán los pasos generales de la clasificación supervisada aplicándolos y ajustándolos a las herramientas de las que se dispone (ilustración 16):

- **Representación:** Los documentos que componen los datos a clasificar deben ser transformados a un formato compresible para el sistema de clasificación a utilizar. La extracción de los datos del conjunto de entrenamiento (parsing)se realizará con la ayuda de las APIs de análisis de las que dispone Tika.

- **Clasificación:**

  - **Entrenamiento:** permite obtener la descripción de las categorías con ayuda de la colección de documentos previamente clasificados. Lo que se denomina "modelo". Creación del modelo se realiza en dos pasos: análisis y encode, del análisis se encarga Lucene. que será específico para el lenguaje en castellano.

El encode, consiste en la obtención del vector de codificación, para este caso Mahout recomienda realizarlo a través de una función hash, midiendo la frecuencia de aparición de una palabra en el texto.

- **Test:** una vez obtenido el modelo, se podrá predecir las categorías de los documentos por clasificar. Algoritmo SGD.

- **Evaluación:** Una vez realizada la clasificación se realizara un análisis de las clasificaciones obtenidas, para poder evaluar la calidad de la clasificación. Las medidas de valuación de las que se dispondrá serán la media y loglikelihood.

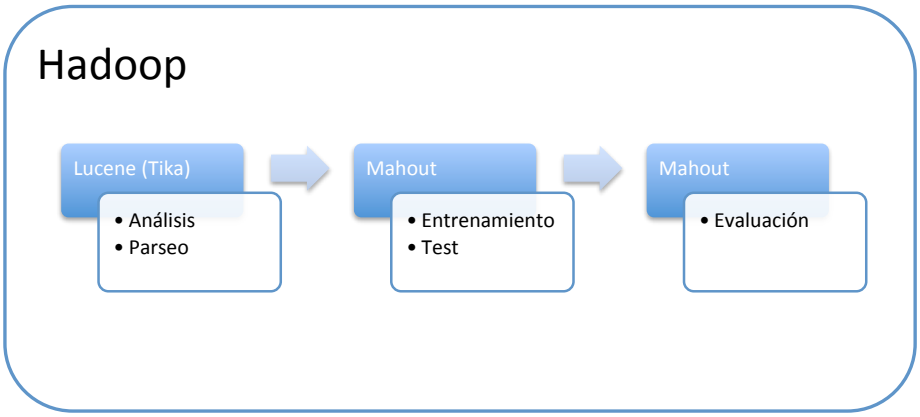


Ilustración 16- Implementación proporcionada por ASF de las fases del proceso de clasificación supervisada.

2.3.Propuesta

2.3.1.Introducción

Como se ha visto en el punto de implementación, la creación del modelo depende de la elección del algoritmo. Por lo tanto en primer lugar seleccionaremos el algoritmo de clasificación, y con sus características se verá la forma en que se debe realizar las fases correspondientes a la clasificación, siguiendo el proceso de clasificación supervisada:

- Representación: Como se realiza la transformación de los ejemplos de los que se dispone para que sean comprensibles para el sistema de clasificación.

- Clasificación:

- Entrenamiento: Método de obtención de las categorías a partir de los ejemplos y construcción del modelo de entrenamiento.
- Test: Análisis de los documentos seleccionados a tal fin. Metodología utilizada.

- Evaluación: Una vez realizada la clasificación se realizará un análisis de la clasificación obtenida basándose en los parámetros de evaluación proporcionados por el clasificador.

2.3.2.Decisión

Una vez mostradas las distintas posibilidades de clasificación que nos proporciona la plataforma Hadoop, se debe seleccionar un algoritmo de clasificación.

Basándonos en los datos anteriores:

- Tamaño del modelo de entrenamiento (disponemos de un máximo de 5.000 ejemplos de entrenamiento).
- Tipo de documentos a clasificar (texto plano).
- Datos continuos.
- Recomendaciones de Mahout.

Podemos concluir que el algoritmo que mejor se adapta a las necesidades de nuestro caso es el SGD, ya que se recomienda su utilización para modelos de poca envergadura y variables continuas.

2.3.2.1.Métodos de gradiente descendente (SGD)

Este método resuelve la regresión lógica a través del gradiente escolástico descendente.

Un clasificador de regresión logística utiliza una combinación de los valores de entrada ajustados al rango (0,1) y utilizando la función $\mathbf{1}/(\mathbf{1}+e^{-x})$. Los resultados obtenidos de un modelo de regresión se pueden interpretar como una estimación de probabilidad. Los coeficientes de ponderación utilizados se pueden calcular de manera incremental independientemente de las dimensiones del vector. Por ello la regresión logística es una opción muy popular en el aprendizaje secuencial. La regresión logística trabaja con datos numéricos, por lo que los valores de texto, palabras y categorías deben de ser codificados en forma vectorial.

SGD: El Este método consiste en seguir el gradiente descendiente del espacio de error del sistema. Con cada instancia de ejemplo, el sistema calcula el error cometido en la predicción usando una función de pérdida.

El uso de este algoritmo se basa en utilizar cada ejemplo de entrenamiento para ajustar el modelo para dar una respuesta más concreta.

Algoritmo	Descripción	Caso de uso
Regresión logística, resuelta por gradiente estocástico	Clasificador brillante, rápido, simple y secuencial, capaz de	Recomienda publicidad a los

descendiente (SGD)	aprendizaje on-line en entornos exigentes	usuarios, clasifica texto en categorías

Ilustración 17- Características del algoritmo seleccionado (SGD)

Este método define una función E(W) que proporciona el error que comete la red en función del conjunto de pesos sinápticos W. El objetivo del aprendizaje será encontrar la configuración de pesos que corresponda al mínimo global de la función de error, aunque en muchos casos es suficiente encontrar un mínimo local lo suficientemente bueno.

El principio general del método es el siguiente: dado un conjunto de pesos W(0) para el instante de tiempo t=0, se calcula la dirección de máxima variación del error. La dirección de máximo crecimiento de la función E(W) en W(0) viene dado por el gradiente ∇E(W). Luego, se actualizan los pesos siguiendo el sentido contrario al indicado por el gradiente ∇E(W), dirección que indica el sentido de máximo decrecimiento. De este modo se va produciendo un descenso por la superficie de error hasta alcanzar un mínimo local.

$$W(t+1) = W(t) - \alpha \nabla E(W)$$

Donde α indica el tamaño del paso tomado en cada iteración, pudiendo ser diferente para cada peso e idealmente debería ser infinitesimal. El tamaño del paso es un factor importante a la hora de diseñar un método de estas características. Si se toma un paso muy pequeño el proceso de entrenamiento resulta muy lento, mientras que si el tamaño del paso es muy grande se producen oscilaciones en torno al punto mínimo.
Sin embargo, los algoritmos de gradiente descendente poseen dos problemas. Primero, suelen quedar atrapados en mínimos locales, generándose de esta manera estimaciones subóptimas de los pesos. Segundo, suelen ser muy lentos por utilizar pasos infinitesimales para alcanzar la solución.

2.3.3.Representación de datos

2.3.3.1.Introducción

Recordemos que se dispone de un conjunto de ejemplos previamente etiquetados para realizar la clasificación. Estos ejemplos se encuentran en Anexo 2 para su consulta.

En esta fase se realizarán las transformaciones necesarias para convertir los datos, que contienen los ejemplos de los que se dispone, en un formato comprensible para el clasificador.

2.3.3.2.Decisión

Las herramientas elegidas para realizar esta tarea serán Lucene y Tika. Realizando Tika el parseo de los documentos para obtener los datos que contienen y Lucene indexando y analizando los datos. En al Ilustración 18 se puede ver en el paso que se encuentra el proceso de clasificación.

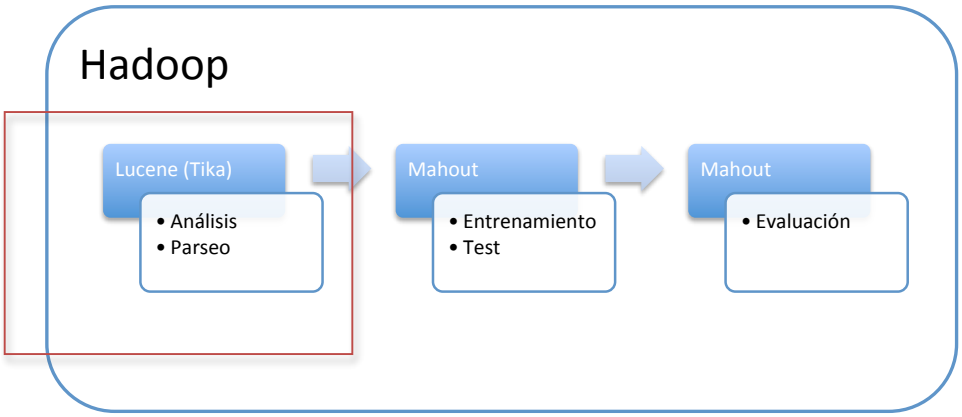


Ilustración 18- Representación de datos

Este proceso será realizado por la herramienta Lucene y Tika, de la siguiente manera:

- Extracción de los datos contenidos en los ejemplos (Tika). Los ficheros del "training-set" (Anexo 2) se encuentran el formato .xml. Por lo que tienen una estructura definida, se obtienen los contenidos de los artículos y la categoría a la que pertenecen. Este proceso se realiza mediante la solución Tika.

- Indexación y búsqueda (Lucene).

    - Análisis de la información separándola en tokens y realizando operaciones sobre ellas como eliminación de artículos o reducción de palabras a su raíz, (algoritmo de stemming de Porter). En el caso a tratar se utilizara el analizador SpanishAnalyzer, el cual divide el texto en tokens basándose en una gramática que reconoce correos, direcciones, acrónimos, caracteres asiáticos, alfanuméricos, etc, pone el testo en minúsculas,

    - Añadir datos al índice. Para ello Lucene usa una estructura de datos conocida como índice inverso que permite búsquedas rápidas por claves.

    - Después de tener los datos indexados se realizaran búsquedas, proceso para buscar palabras en un índice para encontrar documentos en las contengan.

La Ilustración 17 muestra la estructura clásica de un documento xml.

```
<?xml version='1.0' encoding='utf-8'?>

<newsItem xml:lang="es">

<text>

. . .

</text>

 <metadata>

 <codes class="bip:topics:1.0">

 <code code="deporte - fútbol"/>

</codes>

</metadata>

</newsItem>
```

Ilustración 19- Ejemplo de fichero del "training-set"

A continuación se presentan los contenidos de algunos ejemplos, el primero con contenido bien definidos y categorizados en los tres niveles existentes, el segundo vacíos, y el tercero con tres posibles categorizaciones de nivel 2.

La intención de presentar algunos ejemplos es mostrar su contenido y su etiquetado. Se puede comprobar la variedad de textos y calidad de los mismos, así como los distintos niveles de clasificación que se han realizado. Para mas información consultar el Anexo 2.

```
<?xml version='1.0' encoding='utf-8'?>
<newsItem xml:lang="es">
 <text>29 de julio de 2013 | 15:24 pm · Fernando Tablado · Fórmula 1 · Pilotos · Fernando Alonso · Ferrari · Red Bull
El Gran Premio de Hungría 2013 ha servido para barias cosas: confirmar que Mercedes y Hamilton pueden ser una alternativa al título; que Sebastian Vettel lo tiene todo controlado hasta cuando van mal las cosas y que Ferrari se ha dormido en los laureles. Otra vez. El reflejo de la desilusión y la impotencia de Ferrari era Fernando Alonso quien a final de carrera se mostraba más desencantado que nunca en estos últimos tres años.
El piloto asturiano es consciente de que es imposible luchar por el Mundial con un coche que a estas alturas solo permite a sus pilotos ser 5º y 8º. Pero en sus palabras tras la carrera había un trasfondo, una especie de mensaje oculto que dejaba entrever que Alonso está cansado de tanta palabrería barata, de tanto apelar al espíritu y a la historia de Ferrari. Él solo quiere un coche más rápido, y Ferrari, anclada en el pasado, no se lo sabe dar.
Rumores sobre una marcha a Red Bull
En los últimos días se ha estado hablando sobre unas posibles conversaciones entre el manager de Fernando Alonso y Red Bull. Evidentemente el entorno del piloto asturiano se ha apresurado a desmentirlo todo, pero ya sabemos cómo es la Fórmula 1. También es cierto que Fernando Alonso ha dicho que Ferrari será su última escudería en la Fórmula 1 , pero no es la primera ni la última vez
```

```
que donde dije digo digo Diego.
Y es que las opiniones sobre el rendimiento de Alonso son unánimes. El piloto asturiano le saca el 120% a un coche muy inferior, con el que en dos temporadas ha logrado tener opciones de ganar el mundial hasta la última carrera. Se podría decir que Alonso le ha dado más a Ferrari que Ferrari a Alonso, a pesar de no haber ganado el título mundial.
Otra cosa sería ponerse a especular sobre el baile de movimientos que provocaría la salida de Fernando Alonso a Red Bull y si realmente es posible que compartiera asiento con Vettel.
Enviar un comentario
</text>
 <metadata>
 <codes class="bip:topics:1.0">
 <code code="deporte - automovilismo - fórmula uno (f1)"/>
 </codes>
 </metadata>

 </newsItem>
```

Ilustración 20- Ejemplo1 de entrenamiento

```
<?xml version='1.0' encoding='utf-8'?>
<newsItem xml:lang="es">
 <text>Callaway se encargará del desarrollo del Corvette C7 GT3
02 de agosto de 2013 | 11:00 CET
</text>
 <metadata>
 <codes class="bip:topics:1.0">

 </codes>
 </metadata>

</newsItem>
```

Ilustración 21- ejemplo2 de entrenamiento

```
<?xml version='1.0' encoding='utf-8'?>
<newsItem xml:lang="es">
 <text>Londres convoca al embajador español para protestar por los retrasos en la verja de Gibraltar
MADRID, 2 Ago. (EUROPA PRESS) -
    El Gobierno británico ha convocado al embajador español en Reino Unido, Federico Trillo, para expresarle su &quot;seria preocupación por los retrasos en la frontera entre Gibraltar y España&quot; y para pedir garantías de que los registros exhaustivos a los vehículos &quot;no se repetirán&quot; este fin de semana, ha informado el 'Foreign Office'.
    En el comunicado, el secretario de Estado británico de Asuntos Exteriores y para la Commonwealth, Hugo Swire, denuncia los &quot;largos retrasos&quot; de &quot;hasta siete horas&quot; que se produjeron en la frontera entre los días 26 y 28 de julio, que se repitieron el día 30, como consecuencia de los registros &quot;totalmente desproporcionados&quot; a los que sometieron las autoridades españolas a los vehículos que entraban y salían de Gibraltar.
    Según Swire, estas &quot;alteraciones&quot; en el flujo aduanero tiene &quot;un impacto directo en la prosperidad y el bienestar de las comunidades de ambos lados de la frontera&quot;. &quot;La posición del Gobierno de Reino Unido es que estos retrasos son injustificados, inaceptables y no tienen lugar en una frontera entre dos socios de la UE&quot;, recalca en el comunicado.
    Los registros han vuelto a producirse este viernes, cuando se han repetido las colas de entre cuatro y cinco horas para salir del Peñón hacia España, ya que en esta ocasión los controles se han hecho a los vehículos que salían de Gibraltar.
GARCÍA-MARGALLO: LOS CONTROLES SON UNA &quot;OBLIGACIÓN&quot; DE ESPAÑA
    El ministro de Asuntos Exteriores de Reino Unido, William Hague, ya telefoneó el pasado domingo por este motivo al jefe de la diplomacia española, José Manuel García-Margallo, que le explicó que España tiene la &quot;obligación&quot; de hacer estos registros.
    Las reclamaciones continuaron los días posteriores y ayer el Gobierno gibraltareño anunció que ha presentado una queja ante la Comisión Europea porque
```

```
considera que las autoridades españolas están provocando &quot;deliberadamente&quot; retrasos en la frontera.
    En declaraciones a Europa Press, el ministro de Exteriores respondió este jueves a las autoridades del Peñón que España está &quot;cumpliendo estrictamente la legislación&quot; con los controles en la verja, para &quot;evitar el contrabando, los tráficos ilícitos y el blanqueo de dinero&quot;.
    &quot;La diferencia es que nosotros estamos cumpliendo estrictamente la legislación y el Gobierno de Gibraltar ha escogido una política de hechos consumados violando la legislación europea, entre ella la legislación medioambiental&quot;, añadió el ministro, en referencia al lanzamiento de bloques de hormigón que las autoridades de Gibraltar llevaron a cabo la semana pasada en aguas próximas al Peñón, en las que faenan pescadores españoles y que el Gobierno ha denunciado ante la Fiscalía de Medio Ambiente.
SEGUNDA VEZ QUE TRILLO ES CONVOCADO POR LONDRES
    Esta es la segunda vez que Federico Trillo es convocado por el Gobierno de Reino Unido, después de que lo hiciera en noviembre de 2012 para expresarle su &quot;preocupación&quot; por las &quot;incursiones provocadoras&quot; que, según alegaba, se habían producido días antes en &quot;aguas territoriales británicas en Gibraltar&quot;.
    Las tensiones diplomáticas han vuelto a estallar después de que la semana pasada empresas contratadas por el Gobierno de Gibraltar lanzaran al mar hasta 70 bloques de hormigón, con los que dice construir un arrecife artificial para proteger la biodiversidad del fondo del mar, pero que los marineros españoles atribuyen a un intento de impedir que faenen en la zona.
Compartir
</text>
<metadata>
<codes class="bip:topics:1.0">
  <code code="política - diplomacia"/><code code="policía y justicia - policía"/><code code="política - gobierno"/>
 </codes>
  </metadata>

</newsItem>
```


Ilustración 22- ejemplo3 de entrenamiento

2.3.4.Clasificación

2.3.4.1.Introducción

Siguiendo el esquema de la clasificación supervisada, en este apartado se tratará el tema del entrenamiento y test (como muestra la Ilustración). Se realizará un análisis del método empleado por el algoritmo SGD para realizar la clasificación.

- Entrenamiento, obtención de categorías y creación del modelo.
- Test, creación del conjunto de test y validación.

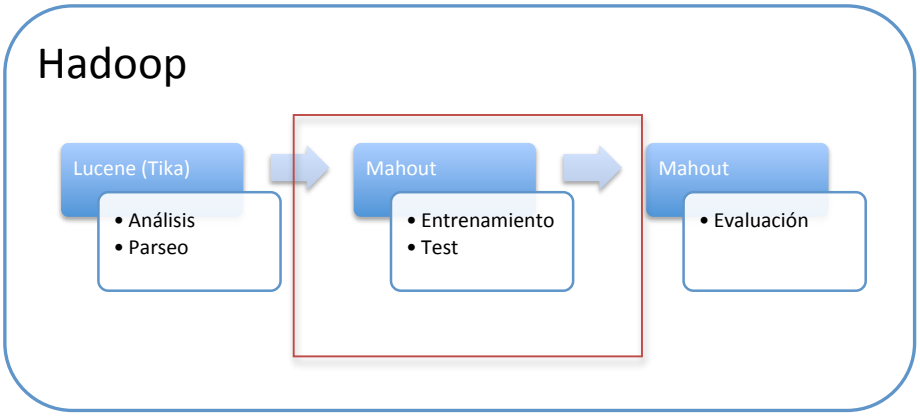


Ilustración 23- Clasificación

Recordemos los pasos que realiza Mahout para realizar la clasificación con el algoritmo SGD:

- Vectorización: Función hash.
- Método de clasificación: CrossFoldLearner.
- Medidas que se ajustan al método de clasificación: Percent correct, Log Likelihood.

2.3.4.2.Vectorización

Al haber elegido el algoritmo SGD para la clasificación la vectorización se realizará con una función hash. En este caso es necesario definir el tamaño del vector de antemano. Se ha utilizado un tamaño de 1000.

2.3.4.3.Entrenamiento y test

El método utilizado por Mahout para la clasificación es el conocido como "Crossfold validation", validación cruzada.

2.3.4.3.1.Método de validación cruzada

El método de validación cruzada o Crossfold, consiste en dividir el conjunto de datos en 10 partes, una se reserva para el test y con el resto se obtienen las categorías, se entrena y se crea el modelo. Seguidamente se prueba el modelo con los datos reservados para el test. Este

proceso se realiza 10 veces, y se dará el modelo que presente mejores resultados, en la Ilustración 24 se muestra un esquema de la aplicación del método.

DATOS (100%)	
Conjunto datos de test (10%)	**Conjunto datos de entrenamiento (90%)**
1º partición de datos	2º a la 10º partición de datos
2º partición de datos	1º, 3º a la 10º partición de datos
3º partición de datos	1º, 2º, 4º a la 10º partición de datos
…	…
10º partición de datos	1º a la 9º partición de datos

Ilustración 24- Método de validación cruzada.

2.3.5.Evaluación

2.3.5.1.Introducción

Una vez realizada la clasificación (entrenamiento y test) se deberá analizar los resultados obtenidos, recordemos que esta fase del proceso de clasificación supervisada la realiza Mahout (como muestra la Ilustración 25). Analizaremos los resultados de las medidas de evaluación resultantes del proceso de clasificación.

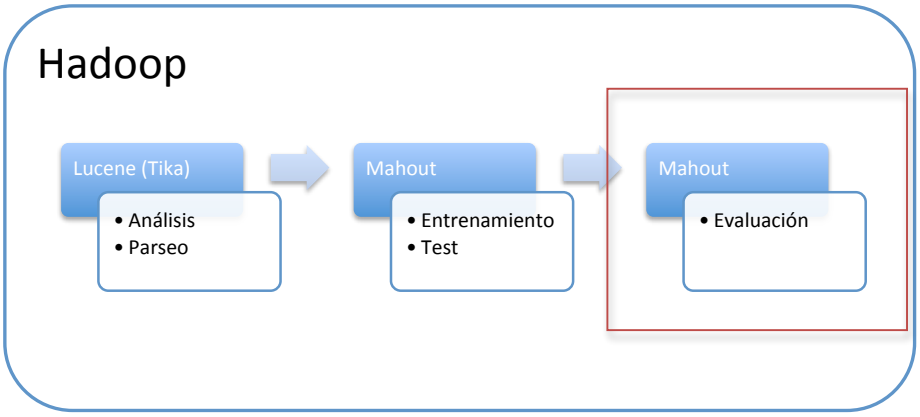


Ilustración 25- Proceso de evaluación

2.3.5.2.Decisión

Las medias de evaluación que se utilizarán serán la precisión (AverageCorrect) y la máxima verosimilitud (Log-likelihood (averagLL)) ya que son las que mejor se ajustan al método de clasificación utilizado en el proceso de clasificación (CrossFoldLearner).

- La precisión indica el porcentaje de documentos bien clasificados.
- Loglikehood es el resultado de maximizar la verosimilitud de los errores de validación cruzada.

A continuación se analizarán los resultados obtenidos en la fase de clasificación.

2.3.5.3.Evaluación de diferentes conjuntos de datos

2.3.5.3.1.Introducción

El proceso de clasificación se ha realizado con varios conjuntos de ejemplos de distinto tamaño y utilizando distintos niveles de categorización:

- 3º niveles de categorización (1400 categorías aprox), 6 conjuntos de 500, 800, 2500, 2750 y 4600 ejemplos respectivamente.
- 1º nivel de categorización (18 categorías como máximo), 18 conjuntos de 500 a 5000 ejemplos.

Los pasos a seguir son los siguientes:

- Dividir el conjunto de ejemplos en un conjunto de entrenamiento (90% de los ejemplos) y un conjunto de test (10% de los ejemplos).
- Obtener las categorías de los ejemplos de entrenamiento.
- Obtención del modelo de entrenamiento.
- Clasificación de los ejemplos de test.
- Resultados de la clasificación.

Este proceso se repetirá 10 veces (CrossFoldLearner) presentando como resultado el mejor de los 10 posibles.

2.3.5.3.2.Evaluación de los resultados obtenidos para la categorización a nivel 3

A continuación se presentan una tabla (Ilistración26) con los datos obtenidos en la evaluación de los cinco conjuntos de ejemplos seleccionados. En la primera columna de la tabla se encuentran el número de ejemplos de cada conjunto utilizados, cada uno de ellos con 500, 800, 2500, 2750 y 4600 ejemplos respectivamente. En la segunda columna se pueden ver las categorías encontradas para los ejemplos proporcionados. La tercera columna proporciona la precisión de la clasificación. En la cuarta columna el Loglikelihood.

Número de ejemplos	categorías	avgCorrect	avgLL
500	123	0,18979592	-0,44266525183
800	158	0,21824908	-0,44919829331
2500	245	0,36448598	-0,40243105123
2750	256	0,378125	-0,38674468659
4600	313	0,39244663	-0,37554631957

Ilustración 26- Tabla de resultados de entrenamiento con los tres niveles de clasificación

En la tabla se puede apreciar que no se han cubierto las mas de 1400 categorías posibles, por lo que se puede considerar que no existe una gran variedad de ejemplos que cubra el mayor número de categorías posibles.

En la gráfica de precisión (Ilustración 27) se pueden ver mas claramente los resultados obtenidos. La precisión no alcanza el 50%, por lo que no se considera un buen modelo.

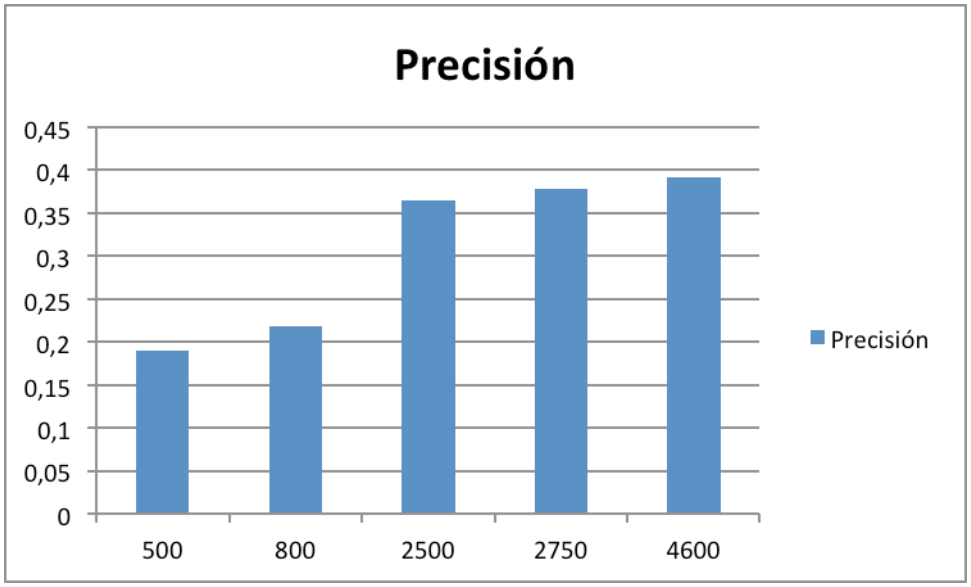


Ilustración 27- Grafica de precisión de los distintos conjuntos de ejemplos para la clasificación en los res niveles de categorización

2.3.5.3.3.Evaluación de los resultados obtenidos para la categorización a nivel 1

Los conjuntos de entrenamiento van de 500 a 5000 elementos y están clasificados en el primer nivel, de forma que se pueden clasificar 18 categorías como máximo.

Una vez dividido el conjunto d ejemplos por el método de clasificación CrossFold, donde se divide el conjunto de ejemplos en 90% de ejemplos para el conjunto de entrenamiento y el 10% de ejemplos para el conjunto de test. Se obtienen los resultados de evaluación, como muestra la Ilustración 28.

En la primera columna el número de ejemplos de cada conjunto seleccionado, en la segunda columna el número de categorías encontradas en los ejemplos de entrenamiento, en la tercera columna (avgCorrect) la precisión de los resultados, en la cuarta columna (avgLL) el Loglikelihood.

Número de ejmplos	categorías	avgCorrect	avgLL
5000	18	0,6556962	-1,627195763
4750	18	0,6482795	-1,617113002
4500	18	0,6472149	-1,601051716
4250	18	0,5763109	-1,855835354
4000	17	0,5994723	-1,691285523
3750	17	0,6009053	-1,681455418
3500	17	0,5287438	-1,859743640
3250	16	0,6047904	-1,640823643
3000	16	0,5751430	-1,803440013
2750	16	0,5940476	-1,699595079
2500	16	0,6175115	-1,873808267
2250	16	0,6203209	-1,684645323

2000	16	0,6063158	-1,607436372
1750	16	0,5649718	-1,695543879
1500	16	0,5347826	-1,734207659
1250	16	0,5781991	-1,751485458
1000	16	0,5076923	-1,951757937
500	15	0,4288660	-2,074127085

Ilustración 28- Resultado de la evaluación para distintos conjuntos de ejemplos y categorizando a nivel 1

De la tabla anterior se puede destacar que salvo el caso de 500 elementos del conjunto de entrenamiento siempre se obtiene una precisión superior al 50% (ver Ilustración 29). Se producen fluctuaciones en la precisión, a medida que cambian el número de elementos a clasificar y las categorías encontradas.

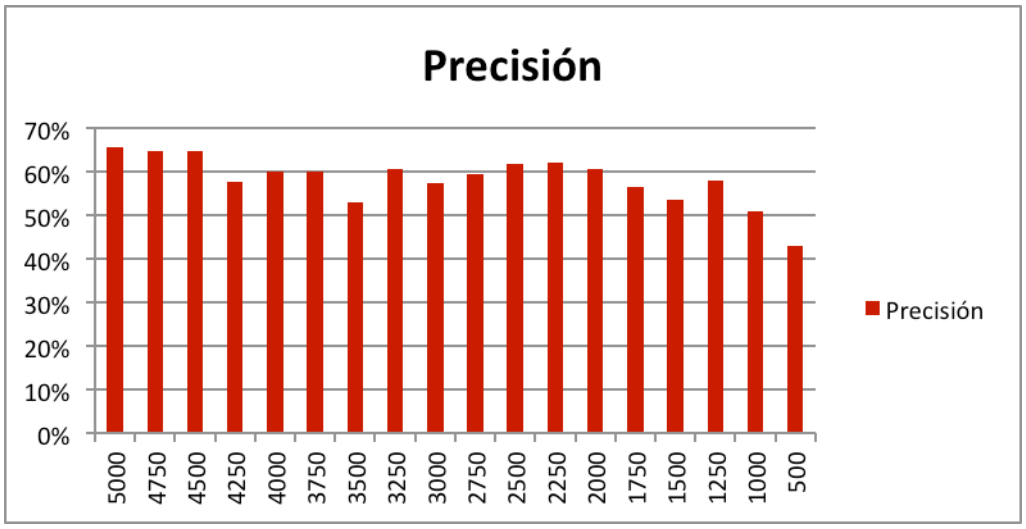


Ilustración 29- Precisión de los distintos conjuntos de ejemplos para la clasificación a nivel 1

A continuación se muestra la precisión por número de categorías encontradas (Ilustración 30, 31 y 32). Se puede ver como la precisión no baja del 60% según baja el número de categorías clasificadas, de forma que mantiene un equilibrio entre número de categorías clasificadas y número de elementos del conjunto de entrenamiento.

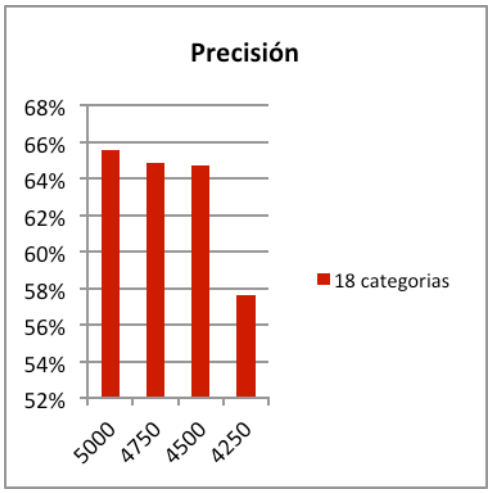


Ilustración 30- Precisión de los conjuntos de ejemplos que proporcionan 18 categorías de clasificación

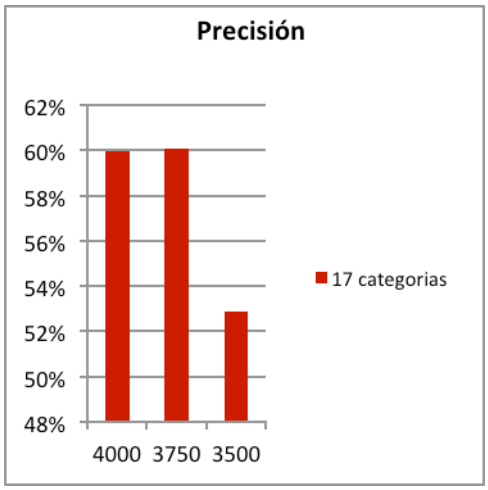


Ilustración 31- Precisión de los conjuntos de ejemplos que proporcionan 17categorías de clasificación

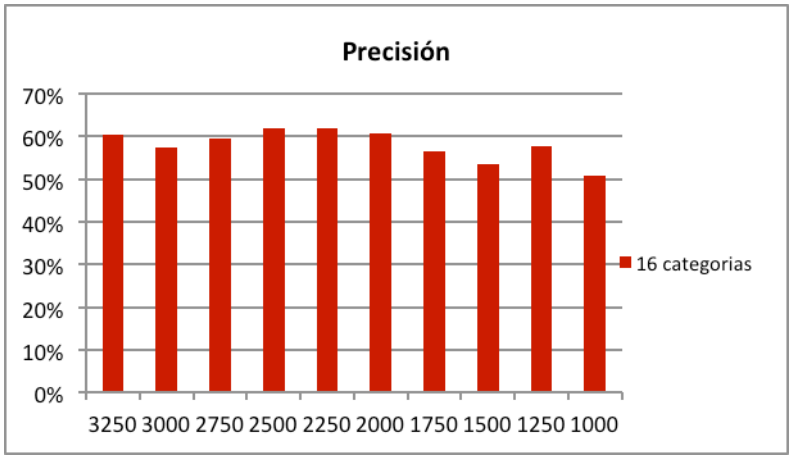


Ilustración 32- Precisión de los conjuntos de ejemplos que proporcionan 16 categorías de clasificación

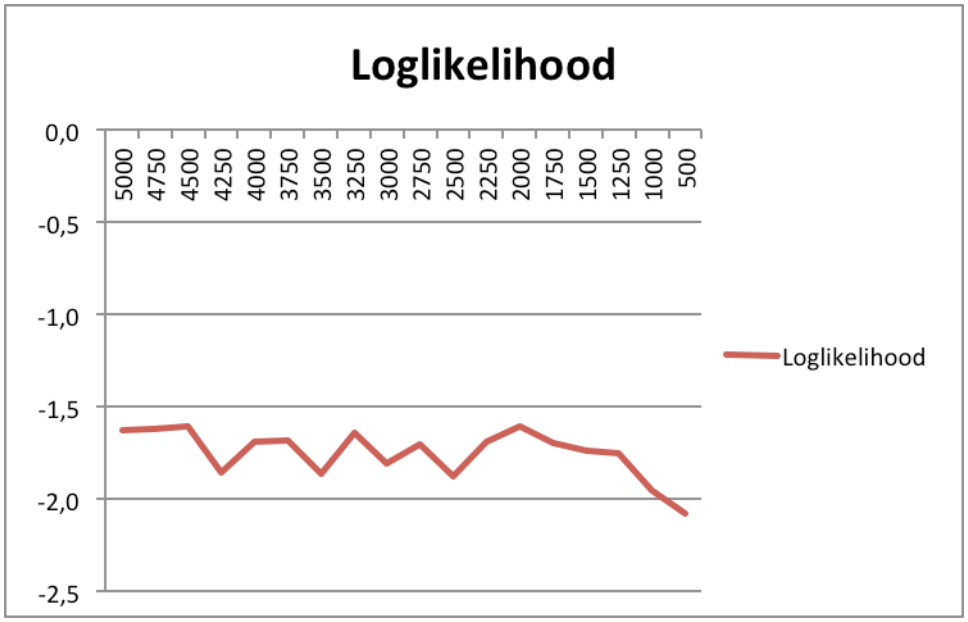


Ilustración 33- Loglikelihood obtenidos en la evaluación de distintos conjuntos de ejemplos a nivel 1

2.3.5.4.Conclusión

En el caso de clasificación basado en los tres niveles de categorización del estándar IPTC, ningún conjunto de ejemplos cubre la totalidad de las categorías, y tampoco proporciona una buena precisión dentro del rango de categorías encontradas, por lo que se considera que no proporciona una buena clasificación.

En el segundo caso, donde la clasificación se realiza a nivel 1 del estándar IPTC, dependiendo del número de ejemplos de los conjuntos seleccionados, se encuentran entre 16 y 18 categorías de las 18 disponibles, en estos tres casos (16, 17 y 18 categorías) la precisión a la hora de clasificar alcanza entre el 60 y 65% de precisión y el Loglikelihood (máxima verosimilitud) sin ser muy bajo mantiene un valor constante entre -1.5 y -2. Por ello consideramos un clasificador aceptable.

Con todo ello tomaremos el modelo formado a partir del conjunto de 5000 ejemplos para implementar la aplicación de clasificación de artículos. Ya que cubre las 18 categorías del primer nivel de categorización del estándar IPTC y proporciona la mayor precisión.

A la hora de presentar los distintos métodos de aprendizaje se dejó en el aire la posibilidad de realizar un aprendizaje semisupervisado. Con los resultados obtenidos no se considera oportuno, ya que si se sumasen al modelo los artículos clasificados solo se ampliaría el error, para realizar una buena ampliación del modelo se tendría que asegurar una mayor precisión en la clasificación de los artículos seleccionados.

2.3.5.5.Análisis de la clasificación de artículos

2.3.5.5.1.Introducción

Una vez realizada la clasificación supervisada, analizada y seleccionado el modelo, se verán unos ejemplos de clasificación de artículos.

Se presentan los resultados de categorización mostrando el porcentaje obtenido por cada categoría en la clasificación del artículo seleccionado.

Se clasifican artículos con distintas características, longitud del artículo, tema a tratar, etc.

2.3.5.5.2.Noticia1

En este ejemplo se clasifica un artículo con un a longitud amplia (Ilustración 34):

Título: “Argentina se paraliza por una huelga general”

Contenido:

Tres de las cinco centrales sindicales de Argentina, las tres opositoras (dos peronistas y otra de izquierda), paralizaron este jueves el país con una huelga general convocada para protestar por una inflación del 32%, subidas salariales que saben a poco y el recorte de subvenciones a las tarifas de gas y agua. No funcionaron los autobuses, los trenes, el metro, ni los aviones. Hasta hubo pocos taxis por temor a los piquetes. Además, los partidos de la izquierda trotskista, que vienen ganando adeptos entre los votantes y dentro de los sindicatos, montaron en toda Argentina unos 5O piquetes en autopistas, carreteras y calles para impedir la circulación de vehículos. Fue la segunda huelga general contra el Gobierno de la peronista Cristina Fernández de Kirchner.

“Pagaste millones para la Repsol y ni un solo peso al trabajador”, cantaban trabajadores de diversos colectivos y

estudiantes universitarios del Partido Obrero en un piquete en una calle paralela a la avenida General Paz, en el límite del municipio de San Martín con la ciudad autónoma de Buenos Aires. Se referían al reciente acuerdo por el que el Gobierno indemnizará con 5.000 millones de dólares a la petrolera española por la expropiación del 51% de YPF. Entre los manifestantes estaba Fortunata Delgado, de 60 años, que migró desde Perú en 2005 porque en su país "no hay trabajo o no te alcanza lo que te pagan". Unos 250.000 peruanos migraron a Argentina entre 2004 y 2012. Fortunata estaba contenta con el kirchnerismo, pero dice que en los últimos dos años aumentaron mucho los precios, en especial los de los alimentos, aparecieron controles cambiarios que limitan las remesas que envía a sus dos nietos en Perú, y el año pasado perdió el empleo que tenía como asistenta. Ahora trabaja de vez en cuando en alguna casa de familia y se alimenta en un comedor de la organización social Corriente Clasista Combativa.

"Es la primera vez [en 11 años de kirchnerismo en el poder] que el aumento salarial no alcanza para cubrir la inflación", se quejaba Julián, de 28 años, obrero de un taller de reparación de trenes situado a pocos metros del aquel piquete. A Julián, un marxista independiente que pronto será padre, le preocupa sobre todo la persecución judicial de los trabajadores que participan en los piquetes.

La huelga fue acatada desde primera hora y las calles permanecieron casi vacías toda la jornada. La gendarmería (policía militarizada) intentó desalojar un piquete en la principal autopista de acceso a Buenos Aires, la Panamericana, y varios manifestantes acabaron heridos y detenidos. El parón se produce en pleno proceso de negociación de subidas salariales con diversos sindicatos. Los maestros, que en la provincia de Buenos Aires pararon durante todo marzo, han conseguido hasta un aumento del 31%. En el sector privado, en cambio, donde las empresas amenazan con despidos ante la caída de la actividad en el primer trimestre del año, sindicatos alineados con el kirchnerismo, como los metalúrgicos, los empleados del comercio minorista y los obreros de la construcción, aceptaron entre un 26% y 29%. Pero la inflación, que hasta 2013 era del 27%, se ha disparado hasta el 32% por la brusca devaluación del peso de enero pasado.

El Gobierno anhela acabar el año con menos de un 25% de inflación, pero todavía no es seguro que los ajustes fiscal y monetario y los acuerdos de precios, vayan a resultar suficientes para lograrlo. Claro que si la actividad económica acaba creciendo solo 0,5%, como prevé el Fondo Monetario Internacional (FMI), es probable que la subida de precios se modere.

"Hay un fuerte acatamiento [de la huelga]", destacó este jueves el sindicalista Luis Barrionuevo. En cambio, el jefe de Gabinete de Ministros de Fernández, Jorge Capitanich, calificó la protesta como "un gran piquete con paro del transporte". El subsecretario general de la presidencia, Gustavo López, opinó que en esas condiciones no se podía medir el nivel real de adhesión a la huelga.

Ilustración 34- Noticia 1

En la Ilustración 35 se pueden ver los valores obtenidos al categorizar. En la primera columna se encuentran las 18 categorías de las que se dispone, en la segunda columna los valores obtenidos en la clasificación y en la tercera columna el porcentaje de los valores de la segunda columna.

categoría	valor	Porcentaje
economía, negocios y finanzas	0.1468281381020209	39,5298%
ciencia y tecnología	0.07106378157128604	19,1321%
deporte	0.06423532855637945	17,2937%
asuntos sociales	0.021411525861385717	5,7645%
arte, cultura y espectáculos	0.01781184248381258	4,7954%
Política	0.012008566282243132	3,2330%
medio ambiente	0.009322576077809158	2,5099%
catástrofes y accidentes	0.008443678704872013	2,2732%
mano de obra	0.006923614578624615	1,8640%
disturbios, conflictos y guerra	0.0058334329092590625	1,5705%
policía y justicia	0.0025432643972304993	0,6847%
interés humano	0.002505818053584245	0,6746%

estilo de vida y tiempo libre	0.0013415176004938285	0,3612%
educación	8.845493868559679E-4	0,2381%
salud	1.4378442784985403E-4	0,0387%
religión y credos	1.1045876815109737E-4	0,0297%
meteorología	2.392970734978001E-5	0,0064%
multitemático	1.2215884564700E-06	0,0003%

Ilustración 35- resultados de clasificación de la noticia1

La Ilustración 36 muestra los valores anteriores, se puede comprobar que la categoría más valorada es la de "economía, negocios y finanzas" y con una gran diferencia respecto a las demás categorías. En la Ilustración 37 se pueden ver los porcentajes de los valores.

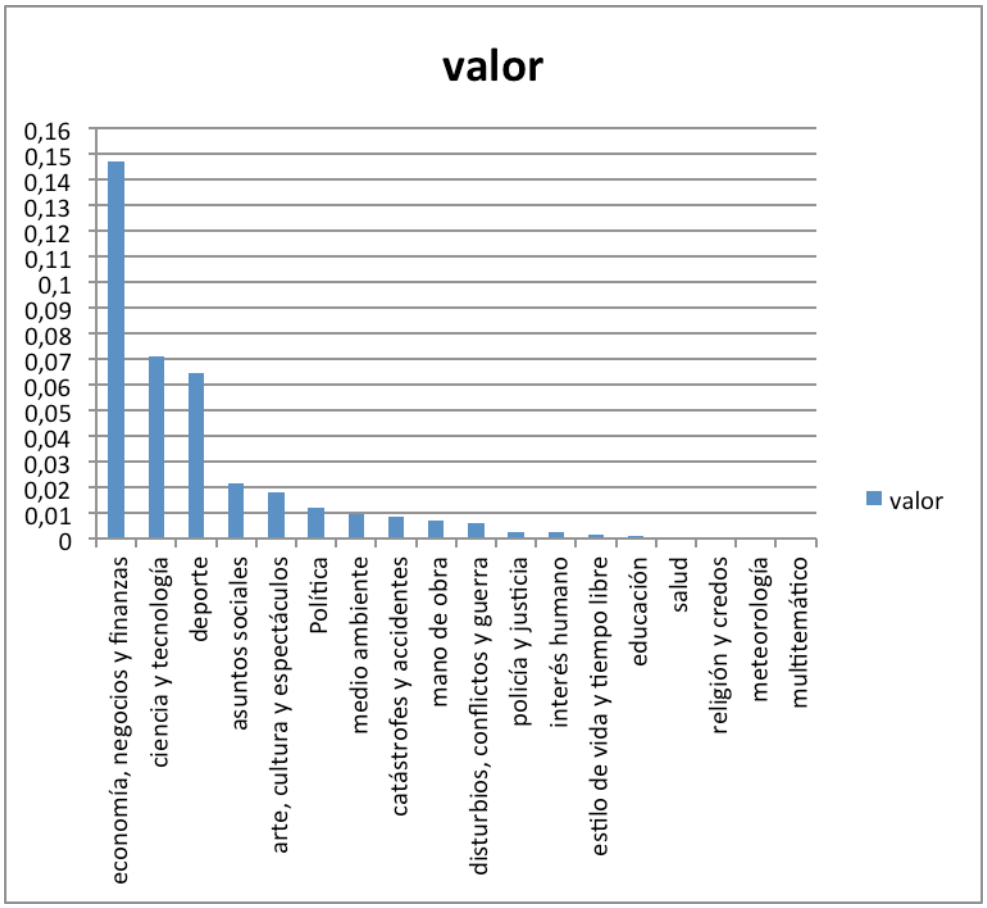


Ilustración 36- Gráfica de los resultados de clasificación de la noticia 1

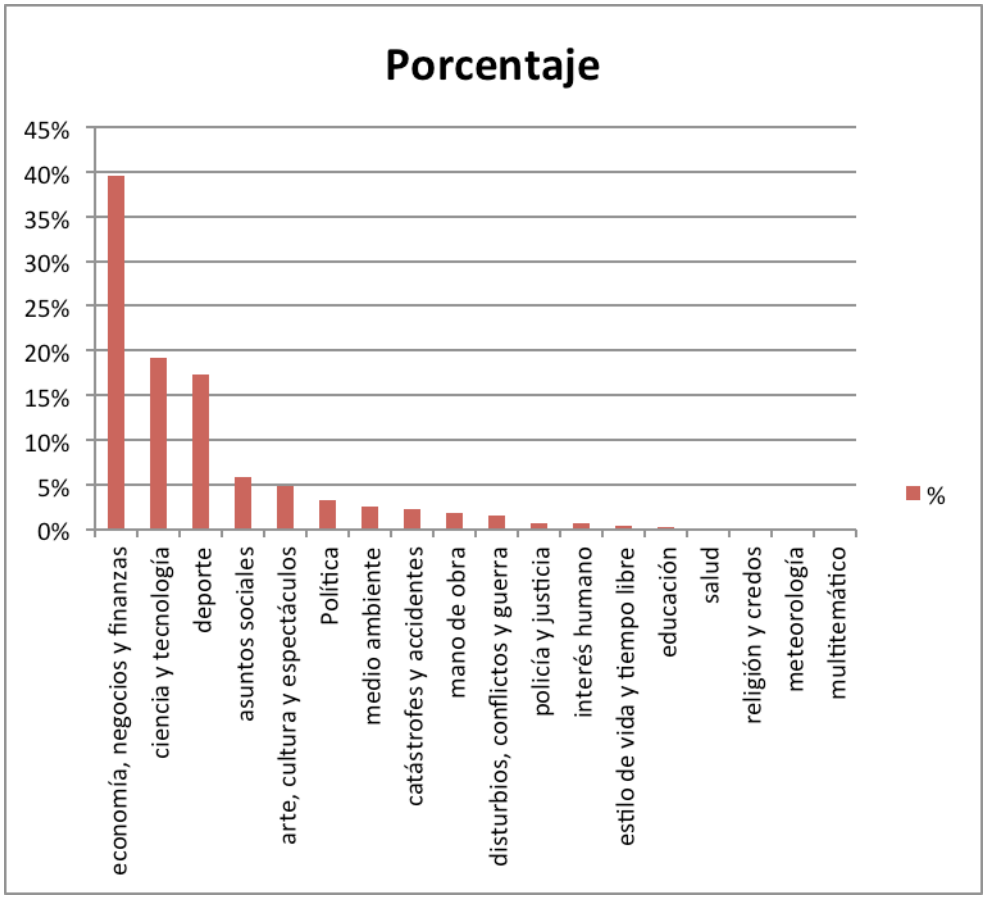


Ilustración 37- Porcentajes de los resultados de clasificación de la noticia 1

2.3.5.5.3.Noticia2

En este ejemplo se presenta la misma noticia pero de dos fuentes diferentes, en el primer caso (Ilustración 38) el contenido del artículo se limita a una línea, en el segundo caso (Ilustración 42) es un artículo amplio.

Noticia2-1

Título: "La inflación cae al 0,1% en marzo, una décima menos de lo esperado

Contenido:

Pese a registrar dos tasas negativas en los últimos meses, el Gobierno ha descartado que la economía española se encuentre en deflación. Leer

Ilustración 38- Noticia 2-1

En la Ilustración 39 se pueden ver los resultados de la clasificación del artículo de la Ilustración 38, en la primera columna aparecen las 18 categorías posibles, en la segunda columna los valores obtenidos en la clasificación y en la tercera columna el porcentaje de los valores de la segunda columna.

Categoría	Valor	Porcentaje
política	0.08034780866057975	8,62%
mano de obra	0.07410219873498711	7,95%
catástrofes y accidentes	0.06549427838094944	7,03%

medio ambiente	0.06515003823980303	6,99%
ciencia y tecnología	0.06277122500610581	6,74%
economía, negocios y finanzas	0.05936218150889656	6,37%
arte, cultura y espectáculos	0.058833688736275785	6,31%
asuntos sociales	0.05793218951662401	6,22%
disturbios, conflictos y guerra	0.05320594461897303	5,71%
educación	0.04732098591239401	5,08%
interés humano	0.04545785113754336	4,88%
estilo de vida y tiempo libre	0.04529373716498008	4,86%
deporte	0.044896689942869064	4,82%
salud	0.038419210594878926	4,12%
religión y credos	0.03783289402928986	4,06%
policía y justicia	0.03775785521385171	4,05%
meteorología	0.030549334762411687	3,28%
multitemático	0.02715746459873522	2,91%

Ilustración 39- resultados de clasificación de la noticia2-1

Noticia2-2

Título: "El INE corrige su dato adelantado de inflación y rebaja la caída en marzo a un 0,1%"

Contenido:

EFE / VÍDEO: EP

- El Índice de Precios de Consumo vuelve a tasas negativas, después de tres meses positivos y uno plano.
- Los grupos con mayor influencia en la variación fueron el de los alimentos y bebidas no alcohólicas y el de ocio y cultura.
- Subieron los precios de la carne de ave, frente a la bajada que se produjo en 2013.

El **Índice de Precios de Consumo (IPC)** registró de nuevo **tasas negativas** en marzo al caer el 0,1% —después de tres meses positivos y uno plano, el 0,0% de febrero—, según el Instituto Nacional de Estadística (INE).

Esta tasa **no se corresponde con el índice adelantado** por el mismo organismo el pasado 28 de marzo, cuando situó la caída en el 0,2%.

Según el INE, los grupos con mayor influencia en la variación negativa de marzo fueron el de **alimentos y bebidas no alcohólicas** (cuya tasa disminuye seis décimas, hasta el 0,5%) y el de **ocio y cultura** (con un descenso anual de casi un punto y medio hasta una tasa negativa del 2,3%). Sin embargo, subieron los precios de la carne de ave, frente a la bajada que se produjo en 2013.

En el caso de los **transportes** se produjo una tasa negativa del 1,3%, cinco décimas superior a la del mes anterior. En este aumento influyó la subida de los precios de los automóviles y la menor caída de los carburantes y lubricantes respecto al mismo mes del año pasado.

Caída en todas las comunidades, excepto en Andalucía

El IPC descendió en marzo en todas las comunidades autónomas, excepto en **Andalucía**, que registró una tasa negativa del 0,1%, una décima por encima de la del mes pasado.

Las mayores bajadas, con una disminución de sus tasas de tres décimas, se produjeron en **Navarra** (con -0,7%) y el **País Vasco** (0,2%).

Crecimiento mensual

Frente a la caída del IPC interanual, **el mensual se incrementó el 0,2%**, especialmente debido a la evolución positiva del vestido y el calzado (4,2%); los hoteles, cafés y restaurantes (0,3%), y el ocio y la cultura (0,3%).

La tasa de **variación anual de la inflación subyacente** (índice general sin alimentos no elaborados ni productos energéticos) descendió hasta el 0,0%.

El **índice de precios de consumo armonizado (IPCA)** registró en marzo una tasa negativa del 0,2%, frente a la positiva de febrero, del 0,1%.

El **índice de precios de consumo a impuestos constantes (IPC-IC)** se situó en una tasa negativa del 0,2%.

Ilustración 40- Noticia 2-2

En la Ilustración 41 se pueden ver los resultados de la clasificación del artículo anterior (Ilustración 40), en la primera columna aparecen las 18 categorías posibles, en la segunda columna los valores obtenidos en la clasificación y en la tercera columna el porcentaje de los valores de la segunda columna.

Categoría	Valor	Porcentaje
política	0.07732658169022592	34,11%
arte, cultura y espectáculos	0.06548890390865807	28,89%
ciencia y tecnología	0.04180616367806744	18,44%
catástrofes y accidentes	0.010045529045725882	4,43%
medio ambiente	0.007989238818263529	3,52%
economía, negocios y finanzas	0.006874791577523755	3,03%
asuntos sociales	0.005659429803967839	2,50%
mano de obra	0.005273214670712425	2,33%
disturbios, conflictos y guerra	0.003159978270147366	1,39%
deporte	0.002067246928651972	0,91%
interés humano	4.3125125799823574E-4	0,19%
educación	2.081393448577838E-4	0,09%
policía y justicia	1.7765514495661434E-4	0,08%
estilo de vida y tiempo libre	1.4909365935042275E-4	0,07%
salud	2.903122866165965E-5	0,01%
religión y credos	1.4736736982592746E-5	0,01%
meteorología	2.2301663456108412E-6	0,00%
multitemático	1.45608511742545E-06	0,00%

Ilustración 41- resultados de clasificación de la noticia2-2

En la Ilustración 42 perteneciente a los valores obtenidos por de las distintas categorías de la Noticia 2-1, siendo la probabilidad de cada categoría bastante similar (Ilustración 44).

En el caso de la Noticia 2-2 los valores de las categorías son mas diferentes, y sus porcentajes varían considerablemente (Ilustración45).

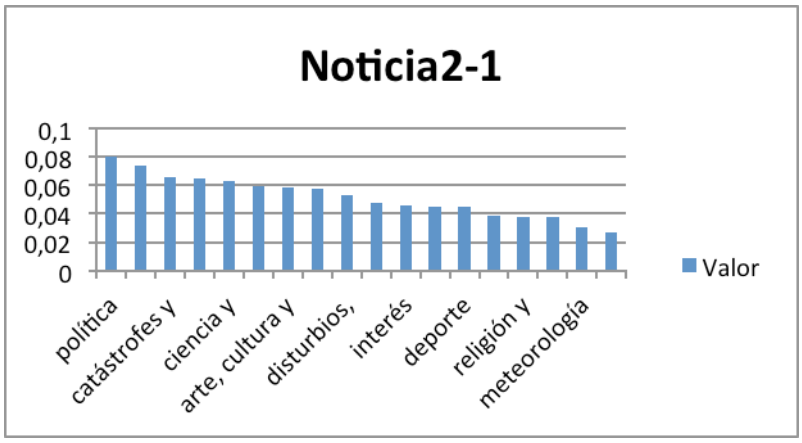


Ilustración 42- Gráfica de los resultados de clasificación de la noticia2-1

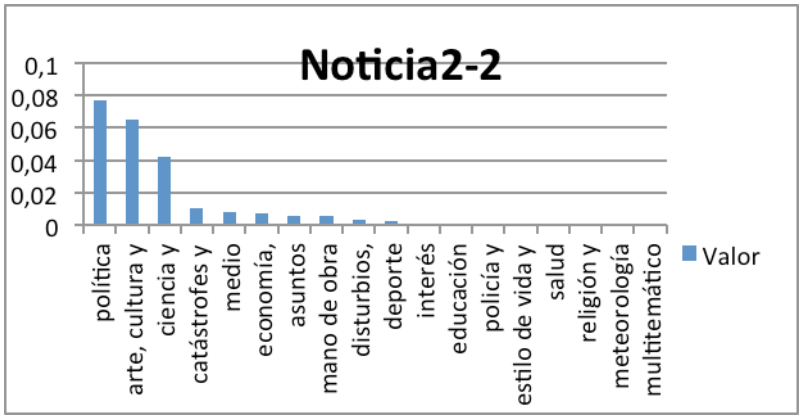


Ilustración 43 Gráfica de los resultados de clasificación de la noticia 2-2

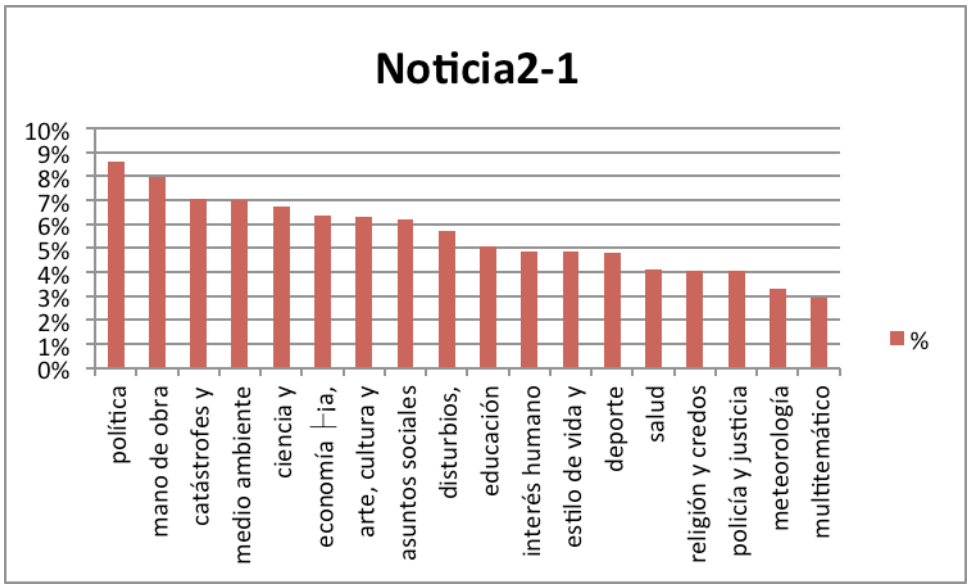


Ilustración 44- Porcentajes de los resultados de clasificación de la noticia 2-1

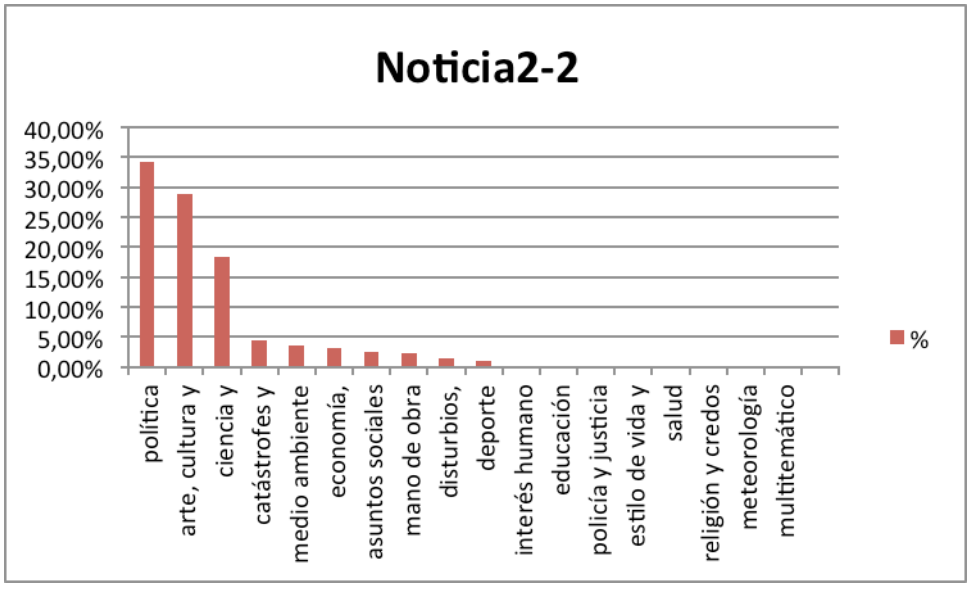


Ilustración 45- Porcentajes de los resultados de clasificación de la noticia 2-2

En los dos casos el resultado de clasificación es el mismo, política. Pero la distribución de las demás categorías es distinta(Ilustración47):

Clasificación	Noticia2-1	Noticia2-2
1º	política	política
2º	mano de obra	arte, cultura y espectáculos
3º	catástrofes y accidentes	ciencia y tecnología
4º	medio ambiente	catástrofes y accidentes
5º	ciencia y tecnología	medio ambiente
6º	economía, negocios y finanzas	economía, negocios y finanzas
7º	arte, cultura y espectáculos	asuntos sociales
8º	asuntos sociales	mano de obra
9º	disturbios, conflictos y guerra	disturbios, conflictos y guerra
10º	educación	deporte
11º	interés humano	interés humano
12º	estilo de vida y tiempo libre	educación
13º	deporte	policía y justicia
14º	salud	estilo de vida y tiempo libre
15º	religión y credos	salud
16º	policía y justicia	religión y credos
17º	meteorología	meteorología
18º	multitemático	multitemático

Ilustración 46- Comparación resultados Noticia2-1 y Noticia2-2

2.3.5.5.4.Conclusiones

A la vista de los resultados obtenidos, se puede concluir que la clasificación de artículos no depende en exclusiva del modelo empleado a tal fin, sino también de la calidad de los propios artículos, su tamaño, estilo literario, etc.

3.Servicio

3.1.Introducción

En esta sección se realizará un análisis de la aplicación realizada.

Para ello se divide el trabajo en varias secciones:

- Especificaciones y requisitos: Introducción o punto de partida, se presenta un diagrama de los pasos que realiza la aplicación para la clasificación.
- Desarrollo: Se realiza una introducción a las herramientas utilizadas en la implementación de la aplicación.
- Diseño:
    - Obtención de datos: Definiciones de feed, RSS, lector. Extración de la información contenida en los feed.
    - Arquitectura, implementación.
- Manuales: Como se utiliza la aplicación.

3.2.Especificación y requisitos

Se realizará un ejemplo práctico de aplicación del algoritmo de clasificación, para ello se contará con RSS de distintos medios de difusión y se clasificarán los artículos que contienen, en base a la categorización del estándar IPTC.

Se partirá de los RSS (Feeds de noticias) suministrados por diferentes medios de difusión, se seleccionará un medio de difusión, se analizara su contenido para obtener los dato necesarios para la clasificación, con ayuda de un modelo de entrenamiento creado para las necesidades del trabajo, se realizará la clasificación por categorías de los artículos contenidos en los RSS. Seguidamente se mostrarán las categorías pertenecientes al RSS seleccionado y los artículos ordenados por dichas categorías.

En el siguiente diagrama de flujo se puede ver la evolución del desarrollo:

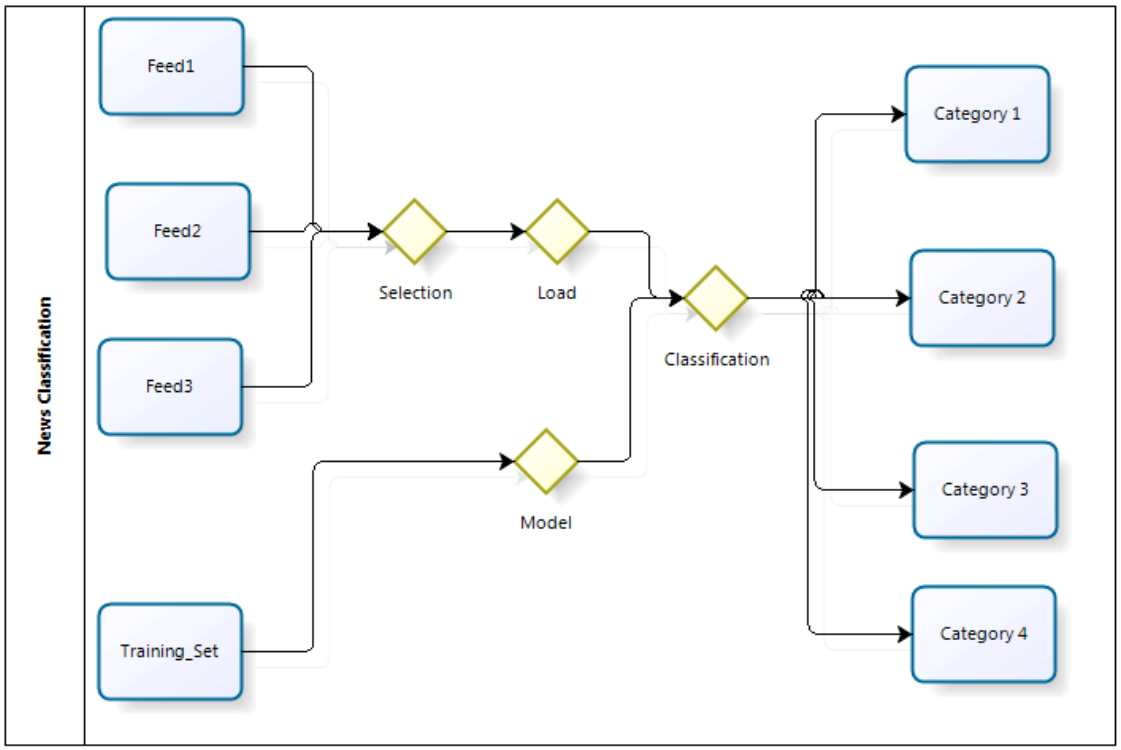


Ilustración 47- Clasificación de noticias

El trabajo se dividirá en cuatro tareas.

1. Obtención de datos: A Partir de ficheros RSS se obtendrán los artículos de prensa.
2. Creación del modelo: con ayuda del conjunto de ejemplos de entrenamiento (o training_set) y su evaluación, se creara un modelo basado en las categorías definidas para realizar la clasificación.
3. Clasificación: Se realizara la clasificación de los artículos con ayuda del modelo y el algoritmo de clasificación seleccionados.
4. Muestra de los resultados de clasificación: Una vez realizada la clasificación de los artículos se mostraran ordenados por sus categorías.

3.3.Desarrollo

Se crea el desarrollo sobre SCALA. Utilizando la aplicación play 2.1. [13]Ello permitirá un desarrollo dinámico y la interconexión de distintos lenguajes de programación, aplicaciones y tecnologías necesarias para las diferentes partes del trabajo.

3.3.1.Scala y play framework 2.1

El nombre de Scala viene de "scalable" y "language", esto indica cual es el propósito de este lenguaje. Scala se trata de un lenguaje de programación multi-paradigma, combina características de los lenguajes funcionales y de los lenguajes orientados a objetos. La implementación actual corre en la máquina virtual de Java y es compatible con las aplicaciones Java existentes.

[13] http://www.playframework.com/documentation/2.1.x/Home

Ilustración 48- http://www.playframework.com/documentation/2.1.x/Home

La plataforma play proporciona una gran ayuda imprescindible a la hora de crear un proyecto de scala.

Play es un framework de desarrollo web tanto para Java como en Scala y comparten gran parte de las API ya que Scala puede importar y utilizar librerías de Java sin problemas.

Una aplicación de Play sigue el patrón de arquitectura para aplicaciones web conocido como MVC (Modelo-Vista-Controlador). De esta forma organiza la aplicación en capas separadas, por una parte la capa modelo y l acapa presentación, esta última se divide a su vez en dos capas, vista y control.

- Modelo: representación de la información específica de la aplicación.
- Vista: Despliegue de la información del modelo de manera que el usuario pueda interactuar con ella, típicamente a través de una interface de usuario (formato web).
- Controlador: Procesa los eventos generados por el usuario. En las aplicaciones web los eventos son, normalmente, http request.

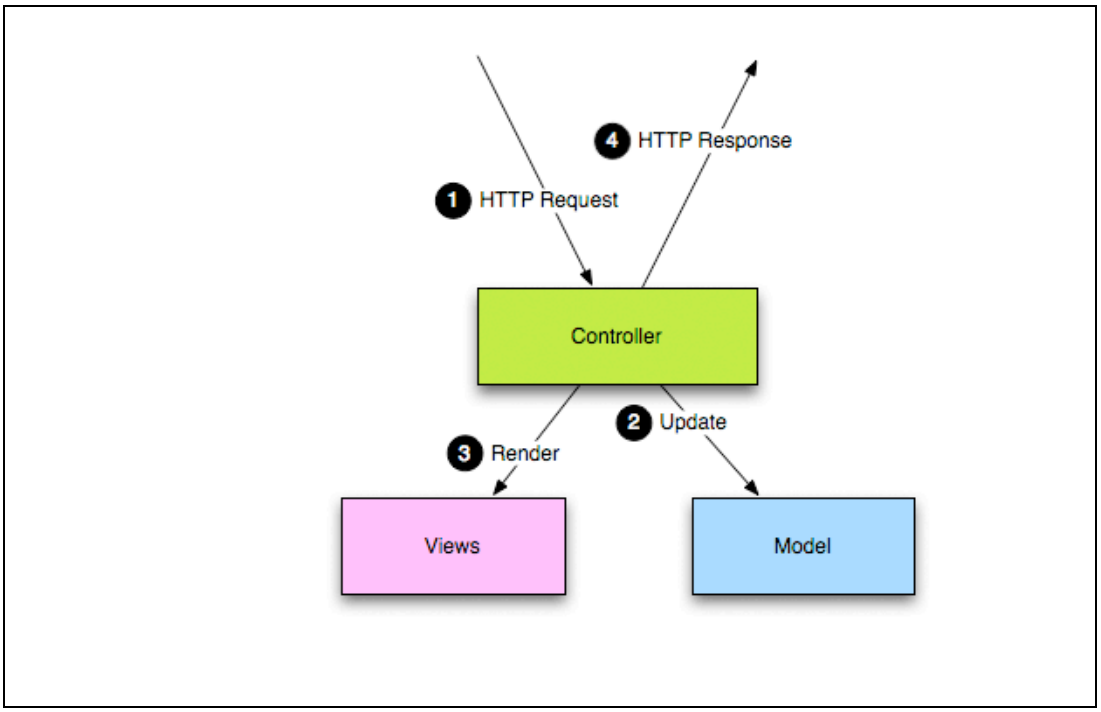


Ilustración 49- El controlador escucha los http request, extrae la información relevante y aplica los cambios al modelo

Al crear una nueva aplicación en scala con play podemos ver su estructura básica:

- app: se encuentra el core de la aplicación; se encuentran los objetos del modelo, controles y vistas. Aquí se encuentran los .scala
- conf: se encuentran las configuraciones, application.conf, routes y archivo de internacionalización.
- project: se encuentran los scripts de creación estos se basan en sbt.
- public: contiene todos los recursos públicos como imágenes, css, javascripts, etc.
- test: se encuentran todos los test que se escriben como especificaciones Specs2

```
PS C:\Users\paula\feed_viewer_heroku> play
[info] Loading project definition from C:\Users\paula\feed_viewer_heroku\project

This project uses Play 2.1.0!
Update the Play sbt-plugin version to 2.2.1 (usually in project/plugins.sbt)

[info] Set current project to feed-viewer (in build file:/C:/Users/paula/feed_viewer_heroku/)

play! 2.1.0 (using Java 1.7.0_51 and Scala 2.10.0), http://www.playframework.org

> Type "help play" or "license" for more information.
> Type "exit" or use Ctrl+D to leave this console.

[feed-viewer] $ run
```

Desde "http://localhost:9000/" se puede Puedes editar tus archivos de Java, guardar los cambios, refrescar el explorar y ver los resultados al instante. No necesitas compilar, desplegar ni reiniciar el servidor.

3.3.2.Heroku

Es un servicio de Hosting basado en la nube de Amazon Web Services, gratuito. Su implementación se realiza a través de Git [14]. La instalación de la app se realiza a través de repositorios.

Inicialmente soportaba solamente el lenguaje de programación Ruby, pero posteriormente se ha extendido el soporte a Java, Node.js, Scala, Clojure y Python y (no documentado) PHP. La base del sistema operativo es Debian o, en la nueva plataforma, el sistema basado en Debian Ubuntu.

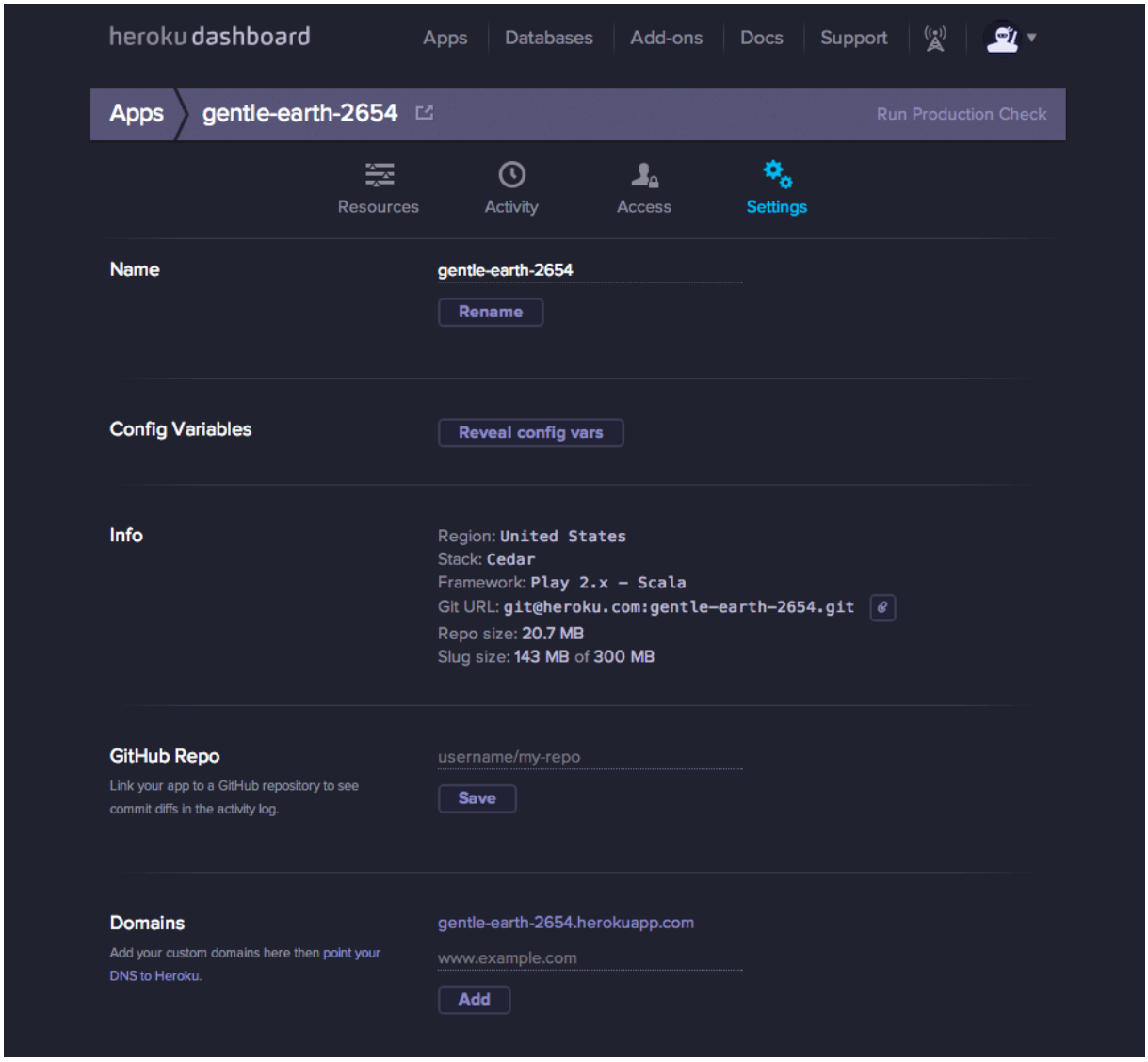


Ilustración 50- http://www.heroku.com

[14] http://git-scm.com/

3.4.Diseño

3.4.1.Introducción

Para la realización de la aplicación se han seguido varios pasos que se comentan a continuación:

- Obtención de los datos: Se recuerda que los artículos a clasificar se encuentran el feed de noticias, por lo que en primer lugar se tendrá que obtener la información contenida en ellos, para ello se introducirán algunos conceptos:
    - Feed
    - RSS
    - Lectores
    - Google API
    - JSON
- Creación y evaluación del modelo.
- Implementación de la aplicación
- Publicación de la aplicación

3.4.2.Obtención de datos

3.4.2.1.Introducción

Para obtener las noticias que se desea clasificar deberemos introducir algunos conceptos.

Las noticias son suministradas por ficheros de datos con formato determinado llamados feeds.

Para acceder a las noticias se deberán descomponer los ficheros obtenidos, para ello nos ayudaremos de herramientas proporcionadas por Google para tal efecto (Google APIs).

A continuación se describen los conceptos más relevantes para la comprensión del proceso de obtención o carga de datos.

3.4.2.2.Feed

Un feed se podría definir como una fuente de información. Son ficheros XML, para homogenizar la estructura de los feeds, se han creado varios estándares, los más destacados son RSS y Atom.

Los Feed RSS es un archivo generado por algún sitio web, contienen una versión específica de la información contenida en esa web y se actualizan frecuentemente, de manera que su contenido cambia sin necesidad de realizar ninguna actuación por parte del usuario del feed. Por ejemplo, si se visita un feed de noticias de un medio de comunicación por la mañana, se

leerán unas noticias, si te visita ese mismo feed por la tarde las noticias estarán actualizadas, sin la necesidad de realizar ningún cambio por parte del lector.

3.4.2.3.RSS

Son las siglas de "Really Simple Syndication". RSS es un formato de archivo basado en XML para que se pueda recibir información actualizada de páginas web en el ordenador o en una página web online, para ello se debe disponer de un lector RSS. Los archivos RSS de reescriben automáticamente cuando se produce una actualización en los contenidos del sitio web al que pertenecen.

Un RSS ofrece un resumen actualizado periódicamente de un determinado contenido web, junto con enlaces a la versión completa de ese contenido, el programa "lector RSS" se encargar de darle formato para su visualización.

Cada elemento de información se llama <ítem>. Contará con tantas <item> como temas tenga a tratar. Cada <ítem> consta de título, resumen, enlace o URL a la página web de origen y otros campos como fecha, autor, descripción, etc.

Estructura básica de un fichero RSS:

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
<channel>
 <title>Título del RSS</title>
 <description>Descripción del RSS</description>
 <link>http://www.sitiodelquesedeseapublicar.com/main.html</link>
 <lastBuildDate>Mon, 06 Jan 2013 00:01:00 +0000 </lastBuildDate>
 <pubDate>Mon, 06 Jan 2013 16:20:00 +0000 </pubDate>
 <ttl>1800</ttl>

 <item>
  <title>Entrada dentro del RSS</title>
  <description>Descripción de la entrada</description>
  <link>http://www.sitiodelquesedeseapublicar.com/enero-2013.html</link>
  <guid>clave única</guid>
  <pubDate>Mon, 06 Jan 2013 17:20:00 +0000 </pubDate>
 </item>

</channel>
</rss>
```

Ilustración 51- Estructura básica de un fichero RSS

3.4.2.4.Lector o agregadores RSS

Es un programa que permite visualizar los contenidos de los RSS.

Actualmente podemos clasificarlos en tres tipos:

1. Aplicaciones de lectura de RSS: Son programas que se instalan en el ordenador. Mientras el programa está en ejecutando, se realizan actualizaciones cada cierto tiempo de las páginas web elegidas por el usuario.
2. Lectores RSS online: Su funcionamiento es similar a los lectores RSS instalados en el ordenador, la diferencia es que su acceso es web, no hace falta instalar ningún programa en el ordenador, pero se deberá dar de alta en el servicio y definir un perfil de usuario con

las preferencias de información. El acceso se realizara mediante el nombre y contraseña proporcionada al darse de alta.

3. Lectores RSS en navegador web o programa de correo electrónico: Se recibirá la información deseada a través de navegadores web o programas de correo electrónico.

En nuestro caso se creará un lector web, al que se podrá acceder sin necesidad de identificación, se podrá seleccionar el medio de prensa de una lista dada y se seleccionara los artículos clasificados que se desee leer por las categorización realizada (estándar IPTC).

Una vez conocida la estructura de los datos de entrada utilizaremos algunas aplicaciones y métodos de obtención de datos específicos para los RSS.

Ya se ha visto en el apartado de implementación una herramienta de parseo (Tika), pero para el caso de los RSS utilizaremos las APIs que nos proporciona Google, ya que dispone de un analizador de feeds.

3.4.2.5.Google API

API de Google (o Google AJAX API) es un conjunto de APIs de JavaScript desarrollado por Google que permite la interacción con los servicios de Google y la integración de RIA (rich Internet applications), multimedia, buscar o añadir contenidos de Internet en aplicaciones web. Se utilizan ampliamente AJAX scripting y se pueden cargar fácilmente usando Google Loader.

Google Loader (o Google AJAX API Loader) es una API de JavaScript que permite a los desarrolladores web cargar fácilmente otras API de JavaScript proporcionadas por Google y otros desarrolladores de bibliotecas populares. Google proporciona un método de JavaScript para cargar un API específico.

Google Feed: Esta aplicación permite descargar y publicar feed, descomponiendo sus contenidos y crear otras APIs.

3.4.2.6.JSON

JSON es un formato de intercambio de datos abierto y basado en texto. Igual que XML, es legible e independiente de la plataforma, además de tener a su disposición una amplia gama de implementaciones. Los datos con formato según el estándar JSON son ligeros y las implementaciones de JavaScript pueden analizarlos sintácticamente con increíble facilidad, lo que lo convierte en el formato ideal de intercambio de datos para aplicaciones web de Ajax. Puesto que JSON es ante todo un formato de datos, no está limitado a las aplicaciones web de Ajax y prácticamente se puede usar en cualquier escenario en que las aplicaciones necesiten intercambiar o almacenar información estructurada como texto.

Como realizar la petición feed con JSON Developer's Guide for the Google Feed API:

Dirección básica de carga: “https://ajax.googleapis.com/ajax/services/feed/load”

Con esta aplicación se conseguirá cargar la información de los feed en nuestra aplicación para ser clasificadas.

Se divide en trabajo en dos partes:

- Creación de los modelos de entrenamiento para su posterior evaluación.
- Implementación de la aplicación con el modelo seleccionado y creado en el paso anterior.

3.4.3.Creación y evaluación de modelos.

3.4.3.1.Introducción

En este apartado se muestra los pasos seguidos para realizar la evaluación de los distintos conjuntos de entrenamiento. Seguidamente se presentan las características de la clasifiación:

- Conjunto de ejemplos en documentos xml.
- Algoritmo de clasificación SGD.
- Idioma seleccionado para el análisis es el español.

3.4.3.2.Evaluación

La evaluación se realizara sobre distintos conjuntos de ejemplos, el proceso será el siguiente:

- Parseo de los ejemplos (formato xlm) y obtención de los datos que contienen, contenido y categoría.
- Clasificación, entrenamiento y test.
- Obtención de datos de evaluación.

Para ello se utilizaran las librerías que proporcionan Mahout(encode) y Lucene(parsing).

Se proporcionan los ejemplos ya clasificados, se preparan los datos para su análisis, XMLContentParser () se encarga de parsear los ejemplos de entrada, este parsing lo realiza con la ayuda de Tika.

Como el algoritmo seleccionado es SGD (TextClassifierFactory.ClassifierType.SGD), se crea el vector con un algoritmo hash (HashMap<String, String> ()), el parsing se realiza para texto en español (SpanishAnalyzer (Version.LUCENE_36))). Los datos de la evaluación se almacenarán en una hoja Excel Para su posterior análisis.

Este proceso se realizará para los conjuntos de ejemplos previamente seleccionados.

- Conjuntos de 500, 800, 2500, 2750 y 4600 ejemplos y 3º nivel de categorización. (véase resultados en el apartado 2.2.5.3.1. Evaluación de los resultados obtenidos para la categorización a nivel 3).

- Conjuntos de 500, 1000, 1250, 1500, 1750, 2000, 2250, 2500, 2750, 3000, 3250, 3500, 3750, 4000, 4250, 4500, 4750, 5000 y 1º nivel de categorización. (véase resultados en el apartado 2.2.5.3.2. Evaluación de los resultados obtenidos para la categorización a nivel 1).

En el cuatro siguiente se muestra el código que implementa este proceso.

```
//obtencion de datos a partir de los ejemplos de entrada ya clasificados
List<Content> trainingSet = new FileLoader(new XMLContentParser()).loadFiles(sourcePath);
    logger.info("training set contains " + trainingSet.size() + " + documents");
    // create classifier: SGD, parsing Lucene, encode hash.
    TextClassifier classifier =
        TextClassifierFactory.createTextClassifier(
            TextClassifierFactory.ClassifierType.SGD,
            new HashMap<String, String>(),
            new SpanishAnalyzer(Version.LUCENE_36));
    // train
    System.out.println("evaluation result: " + classifier.evaluate(trainingSet));
```

3.4.4.Implementación de la aplicación (clasificación de noticias)

3.4.4.1.Introducción

En este punto se tratara de dar una visión de la estructura, arquitectura e implementación de la aplicación.

Hay que recordar la estructura de la aplicación en Scala (Ilustración 53) a la hora de crear el proyecto, de esta manera se tendrá dividido en una estructura de árbol. Con la ayuda de Intellij Idea [15]se puede mostrar la estructura correspondiente al proyecto:

[15] http://www.jetbrains.com/idea/whatsnew/

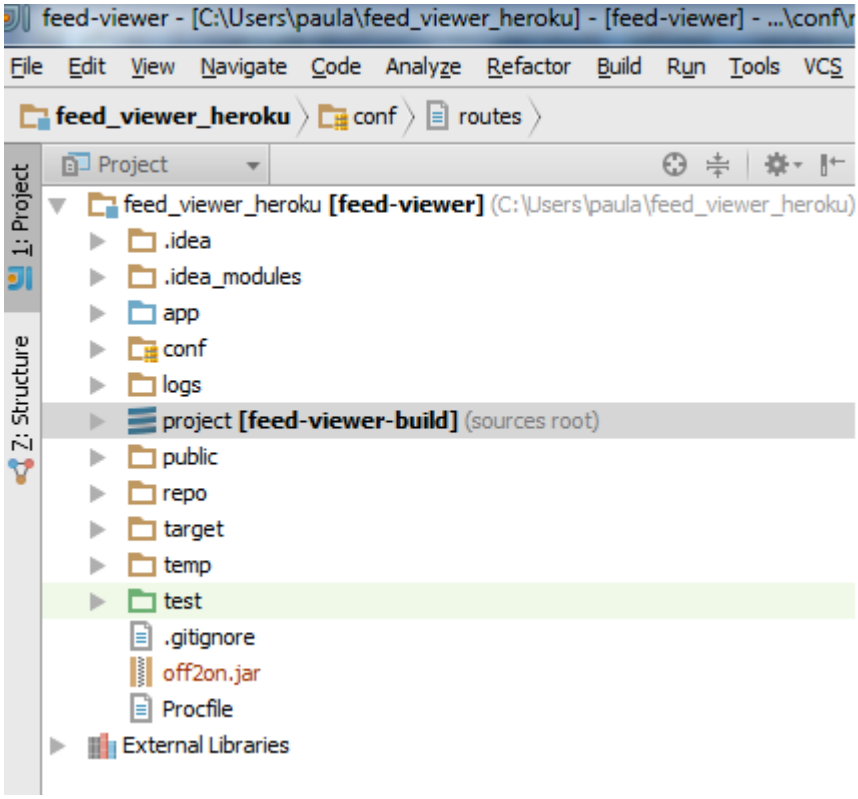


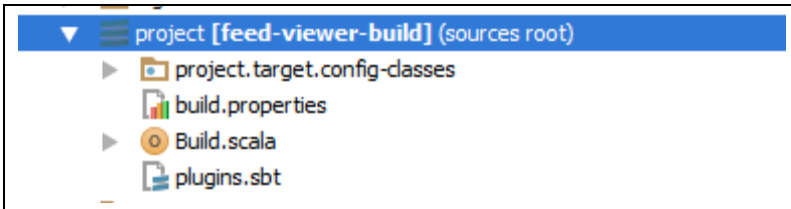
Ilustración 52- Proyecto feed-viewer-heroku por Intellij Idea

Seguidamente se irá explicando detalladamente las ramas del árbol mas destacadas.

- Proyecto
- Repo
- Temp
- Procfile
- External Libraries
- App
- Conf

3.4.4.2.Estructura de aplicación Scala

- **Project**: Construcción del proyecto.



Build.scala: contiene las dependencias

```
"es.ctic.taptap.off2on"%"off2on"%"0.1", //repositorio

"net.databinder.dispatch" %% "dispatch-core" % "0.11.0",  //dispatch library for async-http-client
```

```
"net.liftweb" %% "lift-json" % "2.5.1",   // JSON parsing

"org.apache.hadoop"%"hadoop-core"%"0.20.204.0"
```

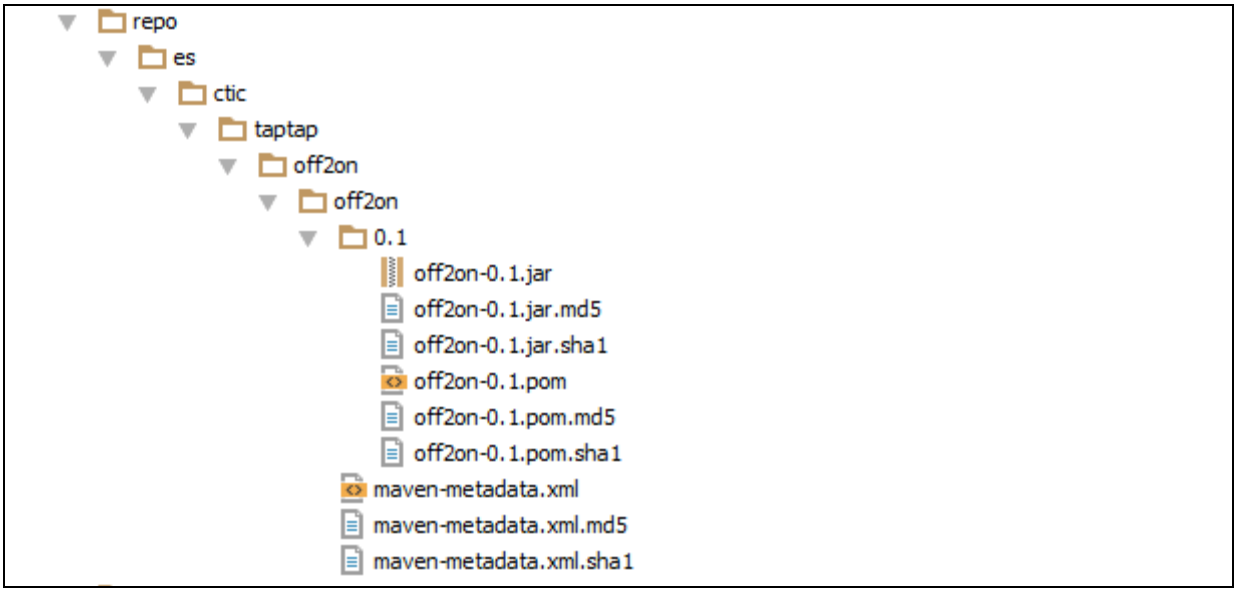
Y donde se encuentra el repositorio.

```
resolvers <+= baseDirectory { base =>
    "local maven repo in project" at ("file:///" + base.getAbsolutePath + "/repo")
  }
```

Plugins.sbt: Tipo de repositorio y versión de Play utilizada.

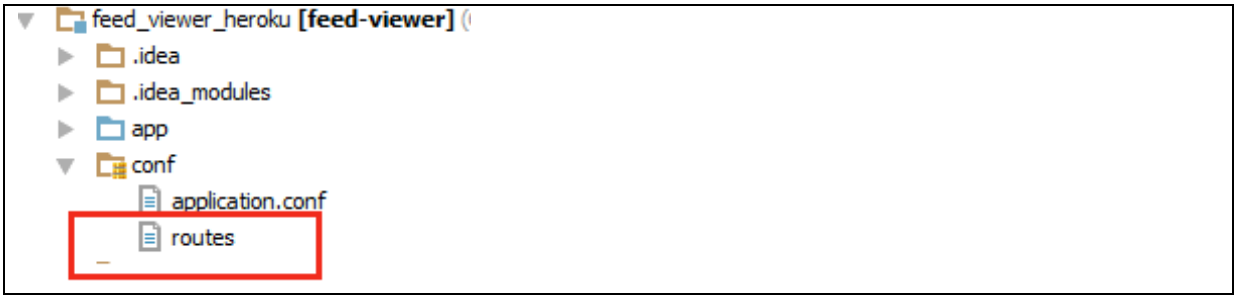
```
// The Typesafe repository
resolvers += "Typesafe repository" at "http://repo.typesafe.com/typesafe/releases/"

// Use the Play sbt plugin for Play projects
addSbtPlugin("play" % "sbt-plugin" % "2.1.0")

addSbtPlugin("net.virtual-void" % "sbt-dependency-graph" % "0.7.1")
```

- **Repo**: contiene las .jar necesarios para la aplicación, procedentes de las librerías de Mahout y Lucene. Las librerías han sido proporcionadas por la fundación CTIC, se han realizado las modificaciones oportunas para adaptarlas al trabajo.



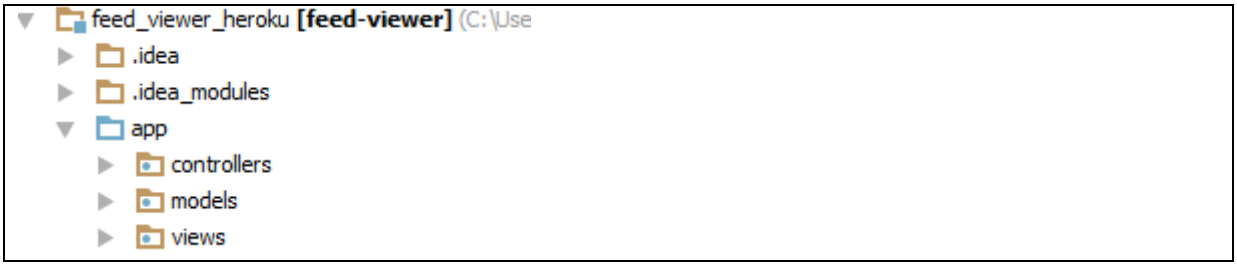
- **Temp**: Almacena el modelo para realizar la clasificación. De los modelos creados para la evaluación se seleccionó el que proporcionaba mejores resultados, el creado a partir del conjunto de 5000 ejemplos.

- **Procfile**: web: target/start –Dhttp.port=$PORT $JAVA_OPTS
- **External Libraries**: Conjunto de librerías necesarias para la correcta ejecución del proyecto.
- **Conf:** Indica las rutas a seguir para la correcta ejecución de la aplicación.



GET	/	controllers.Application.index
GET	/view-all	controllers.Application.view(rss:Option[String])
GET	/category/:name	controllers.Application.articleincategory(name:String)
POST	/article/	controllers.Application.articlecontent

- **App:** Cuerpo de la aplicación, se divide en tres partes, controllers, models y viewer.

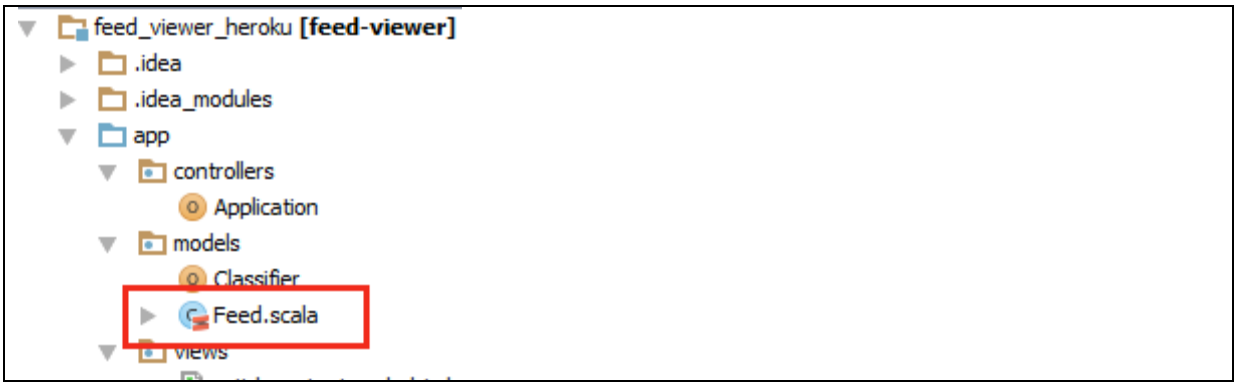


1) Models:
   a) Feed.scala:Obtención de los artículos a clasificar a partir de los Feed de noticias.
   b) Classifier: Clasificación de los artículos según el critero de clasificación de categorías del estandarr IPTC.
2) Controllers:
   a) Application: Realiza el control de las Response/Request HTTP.
3) Views:
   a) Vistas de la aplicación necesarias para la correcta gestión de la aplicación por el usuario.

A continuación se analiza el contenido del directorio App con mas detalle:

3.4.4.2.1.Directorio App

1) Models:
   a) Feed.scala: Obtención de los artículos a clasificar a partir de los Feed de noticias.



Se parte de RSS que se deberán descomponer siguiendo su estructura, se creara una clase Feed con los campos correspondientes a la cabecera del feed y los campos correspondientes a los ítems (artículos):

```
case class Feed(
        feedUrl: String,
        title: String,
        link: String,
        author: String,
        description: String,
        feedType: String,
        entries: List[Entry])
case class Entry(
        title: String,
        link: String,
        author: String,
        publishedDate: String,
        contentSnippet: String,
        content: String,
        categories: List[String])
```

Para ello se necesitara la ayuda de APIs de Google:

```
implicit val formats = net.liftweb.json.DefaultFormats

        val feedApiUrl = "https://ajax.googleapis.com/ajax/services/feed/load"
        val feedApiHost = "ajax.googleapis.com"
        val feedVersion = "1.0"
```

Argumentos utilizados:

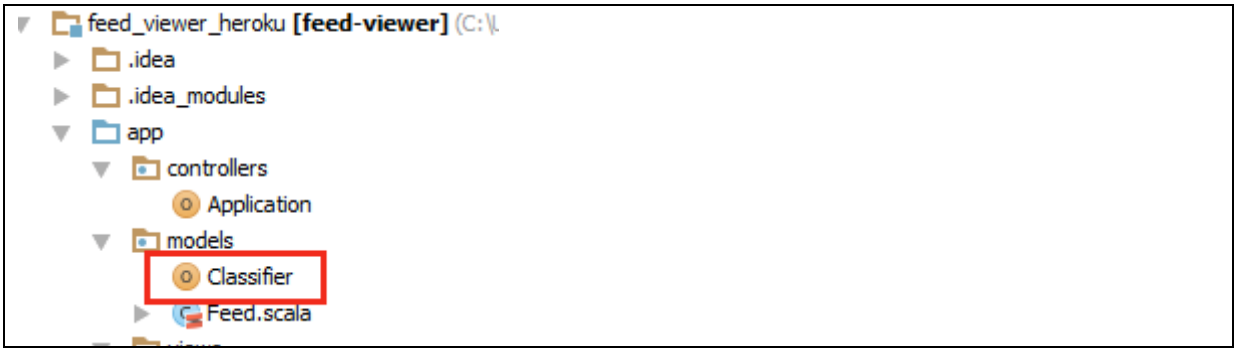
- “q”: Este argumento suministra la consulta, para un feed de carga, que es el caso que se trata, consiste en una URL.
- “v”: Este argumento proporciona el número de versión del protocolo. El único valor válido en este momento es “1.0”.
- “num”: Este argumento proporciona el número de entradas a cargar desde el feed, el valor máximo actualmente es de cien. Si no se especifica, solo se realizaran cuatro entradas de cargas del feed.

```
val numarticle: String = rssUrl.size.toString
val service = url(feedApiUrl)
    .GET
    .addQueryParameter("v", feedVersion)
    .addQueryParameter("q", rssUrl)
    .addQueryParameter("num",numarticle)
    .addHeader("Referer", "http://ctic.es")
```

Se parsea:

```
private def parseDocument(rawResponse: String): ApiResponse = {
      JsonParser.parse(rawResponse).transform {
        case JField("type", x) => JField("feedType", x)

      }.extract[ApiResponse]
}
```

b) Classifier: Clasificación de los artículos según el critero de clasificación de categorías del estandarr IPTC.



Una vez descompuesto el RSS en campos, se deben clasificar los artículos.

El modelo se encuentra en el path:

```
val modelPath: String= "temp/model"
```

Se carga el modelo indicado anteriormente:

```
    classifiertemp.loadModel(new File(modelPath).getCanonicalPath)
```

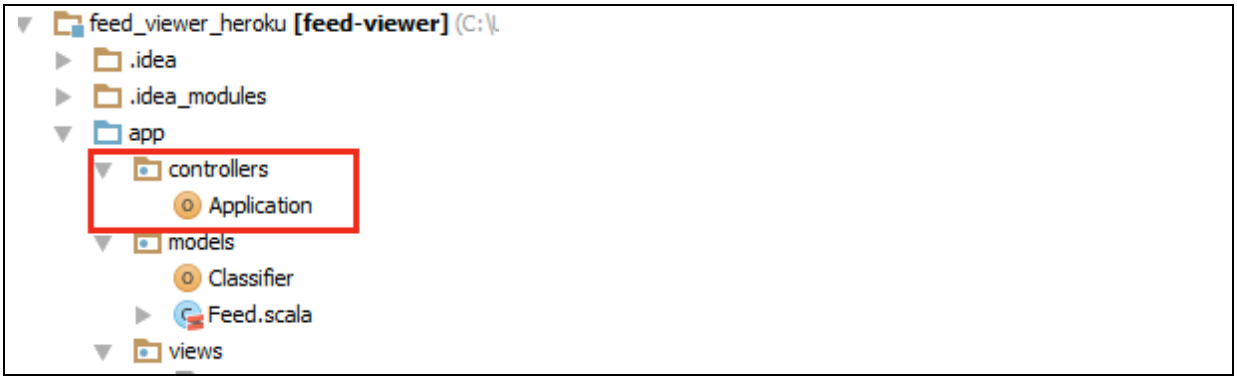
Se realiza la clasificación utilizando el algoritmo SGD y parseando el texto en castellano:

```
val classifiertemp: TextClassifier = TextClassifierFactory.createTextClassifier(
    TextClassifierFactory.ClassifierType.SGD,
    new util.HashMap[String, String],
    new SpanishAnalyzer(Version.LUCENE_36))
```

Seleccionando la mejor categoría, que se encuentra en el primer lugar de la lista de categorías:

```
result=allresult.head

    val category: String = result.getCategory()
```

2) Controllers:
   a) Application: Realiza el control de las Response/Request HTTP.



Presentación inicial:

```
def index = Action {

   Ok(views.html.view(result,None,title))

 }
```

Muestra del contenido de un artículo seleccionado:

```
def articlecontent= Action { implicit request =>

   val formData:Map[String,Seq[Object]] =

request.body.asFormUrlEncoded.getOrElse(Map.empty)

   val inputText: String =

formData.getOrElse("content",Seq("")).apply(0).asInstanceOf[String]

 Ok(views.html.articlecontent(inputText))
```

```
  }
```

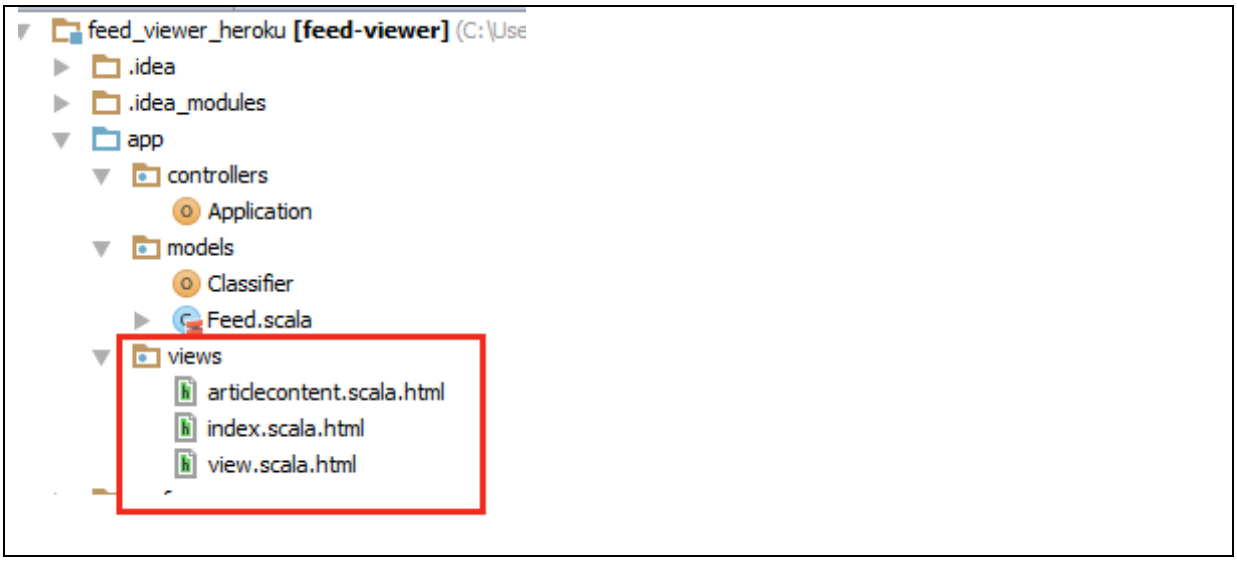
Muestra de los artículos correspondientes a una categoría dada:

```
def articleincategory(category:String)= Action{

   Ok(views.html.view(result,Some(category),title))
  }
```

Clasificación de los artículos contenidos en una Fuente de noticias, muestra las distintas categorías encontradas:

```
def view(rss:Option[String]) = Action {
  //reinicializacion de variables
   categories= List.empty
   result= Map.empty
   categoryselect=None
   //obtencion de la rss o cambio de rss
    val feed = Feed.loadFeed(rss.get)
    //Nombre del medio de comunicación para verlo por pantalla
    title=feed.title

    for(article<-feed.entries)
    {
      //Se clasifica un articulo y se seleciona la primera categoria obtenida
      val category= Classifier.classifier(article.content)

     //Se crea un mapa con las categorias obtenidas y los articulos correspondientes a dichas
categorias
      result+=(category->(article::(result get category getOrElse Nil)))

      categories::=category


    }

    categoryselect=Some(maincategory)

    Ok(views.html.view(result,categoryselect,title))
  }
```

3) Views:
   a) Vistas de la aplicación necesarias para la correcta gestión de la aplicación por el usuario.



3.4.4.3.Publicación

Resulta obvio comentar que al recibir los datos vía web (feed de noticias) se devuelvan los resultados de la clasificación en el mismo medio.

Por ello se ha publicado la aplicación.

A la hora de publicar la aplicación se tuvieron algunos problemas de integración causados por las diferentes versiones empleadas.

El primer problema encontrado ha sido la localización del repositorio correspondiente a las librerías de clasificación, se recordara que estaban implementadas en java. Este repositorio se debe situar dentro de la aplicación para que sea localizada, por otra parte se debe crear el .jar con maven en la versión 1.6.0 de java, ya que de otra manera no será admitida por play framework.

Otro problema encontrado ha sido la compatibilidad de versiones, se debe comprobar la versión de play framework con la que se trabaja, ya que es muy importante a la hora de crear el Procfile, fichero que necesita Heroku para ejecutar la aplicación.

Play framework 2.1..	web: target/start –Dhttp.port=$PORT $JAVA_OPTS
Play framework 2.2..	web: target/universal/stage/bin/myapp –Dhttp.port=$PORT

Una vez solventados se realizó el proceso para publicar la aplicación:

Se instala heroku y se siguen los pasos indicados para subir la aplicación, se realiza en la terminal, en el directorio perteneciente a la aplicación deseada.

Identificándonos en Heroku

- Además de nuestro correo electrónico, necesitamos generar una key ssh para acceder al repositorio
- $ ssh-keygen.exe
    - indicamos la ruta donde se almacenaremos la clave rsa
    - /C/Users/name/.ssh/id_rsa

- introducimos una passphrase
- Agregamos nuestras clave
  - $ heroku keys:add

$ heroku login: Después de autentificar, se generará la clave ssh para acceder al repositorio.

$ heroku créate - -stack cedar: se asigna una dirección donde se ejecutara la aplicación.

$ git init

$ git add

$ git commit –m “init”

$ git push heroku master

$ heroku open

La aplicación se encuentra publicada en :

gentle-earth-2654.herokuapp.com

3.5.Manual

3.5.1.Introducción

La aplicación desarrollada se ejecutara vía web. Por lo que se manejo es sencillo e intuitivo.

En la Ilustración 53 se muestra un esquema de su uso y en el esquema siguiente los pasos a seguir obtener la clasificación deseada:

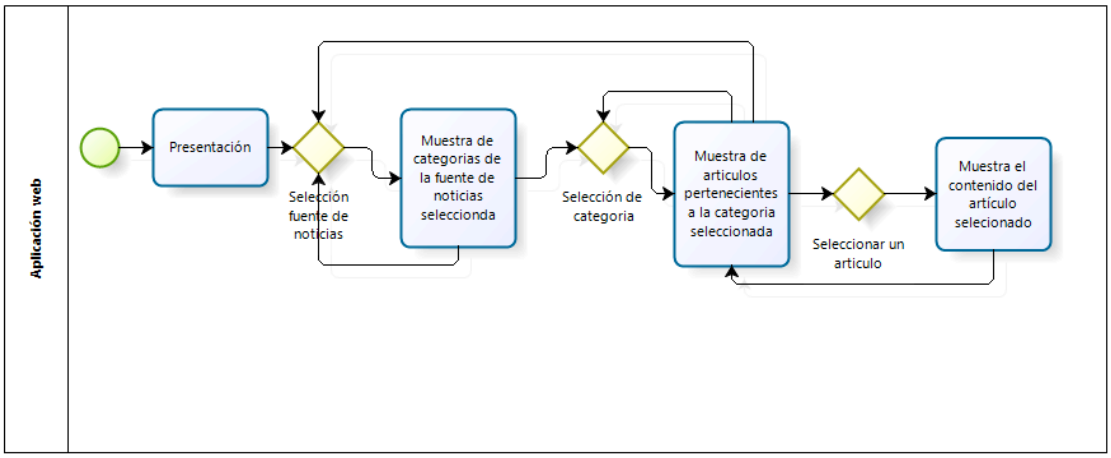


Ilustración 53- Diseño de la aplicación

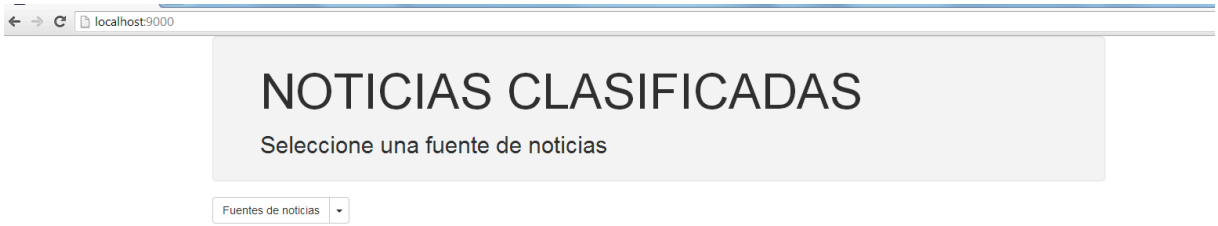
- Para iniciar la aplicación se escribirá en el navegador la dirección: gentle-earth-2654.herokuapp.com
- Se verá la página de inicio, donde se podrá seleccionar el medio de difusión del que se quieren clasificar los artículos.
- En este punto se mostrarán las categorías encontradas.
- Si se selecciona una categoría mostrará los títulos de los artículos que contiene.
- Si se selecciona un artículo mostrara su contenido.

Para crear la aplicación web se ha utilizado layoutit. LayoutIt[16] es un entorno de trabajo que permite crear diseños web de forma gratuita. Creará una plantilla que servirá de base para el desarrollo web.

3.5.2.Presentación

A continuación se presentan capturas de pantalla de la aplicación, mostrando los distintos pasos aseguir.

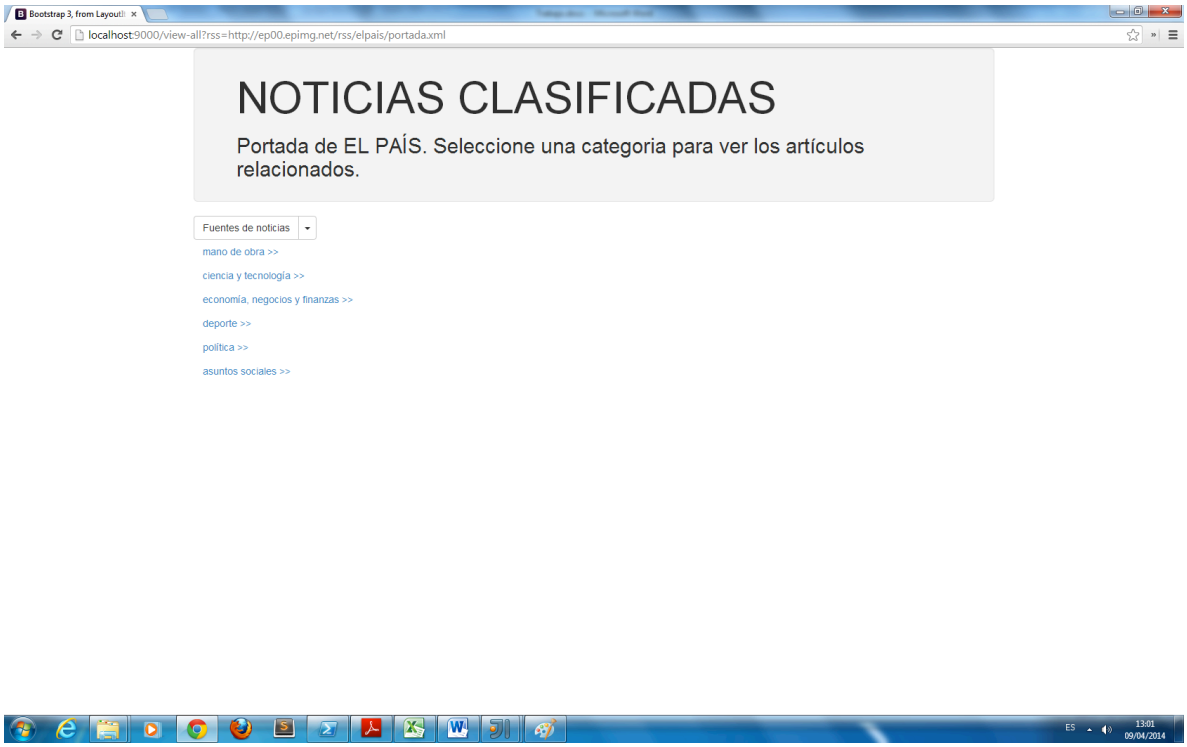
La aplicación muestra una página de presentación, con el título, botón despegable de selección y cuadro explicativo de presentación y funcionamiento.



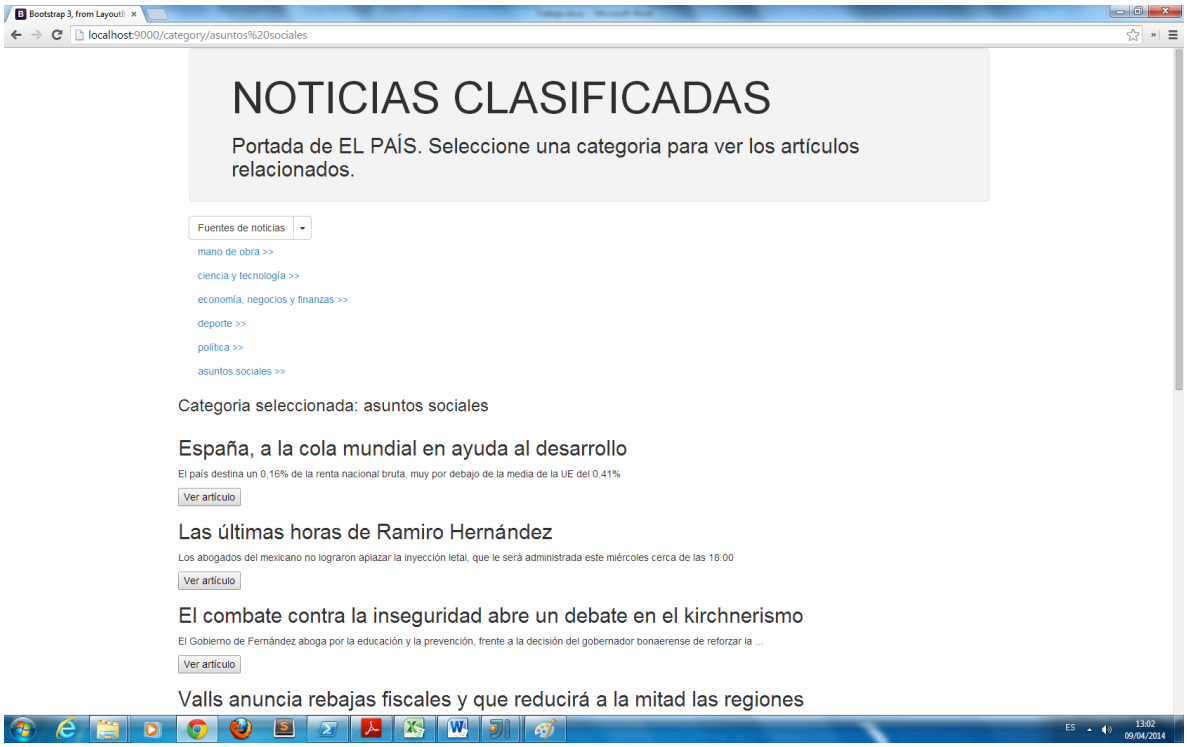
La aplicación permite seleccionar una fuente de noticias, en ese momento se realizara una clasificación de las noticias de la fuente seleccionada, se mostrará un botón por categoría

[16] http://www.layoutit.com/

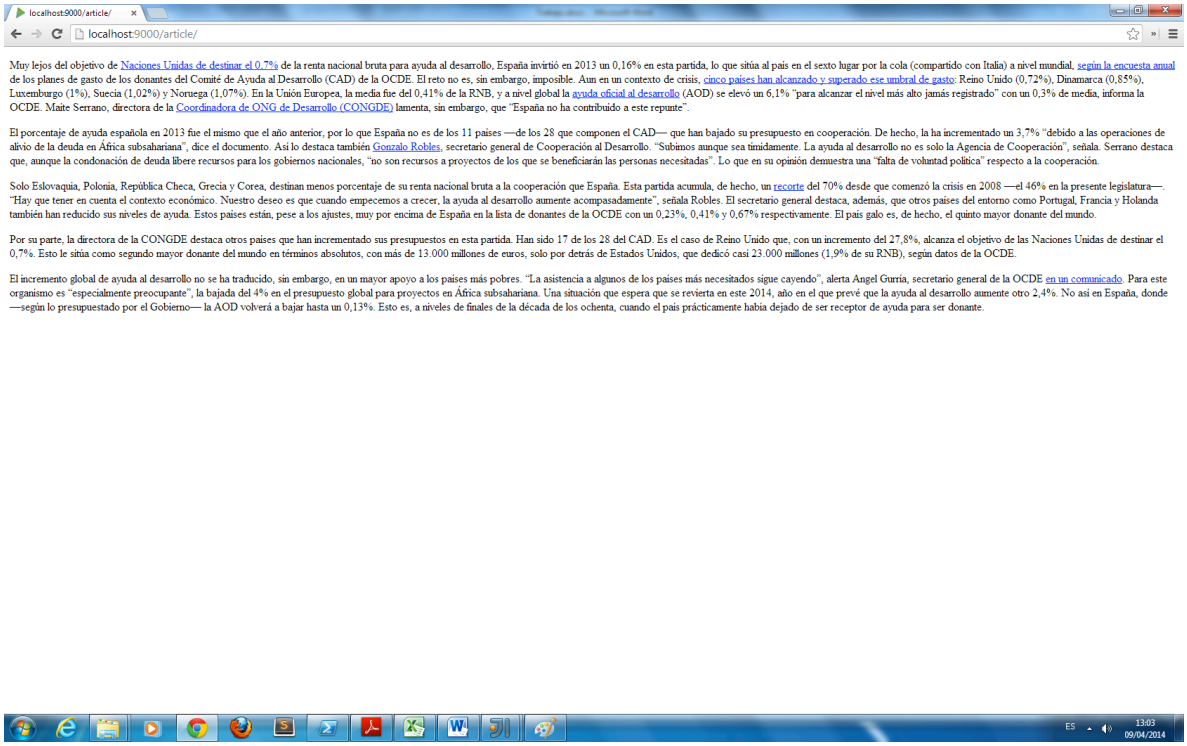
encontrada y un cuadro explicativo de los pasos a seguir.



Si se pulsa el botón de un tema mostrara un cuadro por noticia relacionada, en dicho cuadro aparecerá el título de la noticia y un botón, que si es pulsado se podrá visualizar el contenido de la noticia. Si se quiere ver otros contenidos, se deberá volver atrás. En el momento que se cambie la fuente de noticias se creara la nueva clasificación y se podrá seleccionar de nuevo las noticias deseadas.



Pulsando en el botón ver artículo, se muestra el contenido del artículo.



4.Planificación y presupuesto

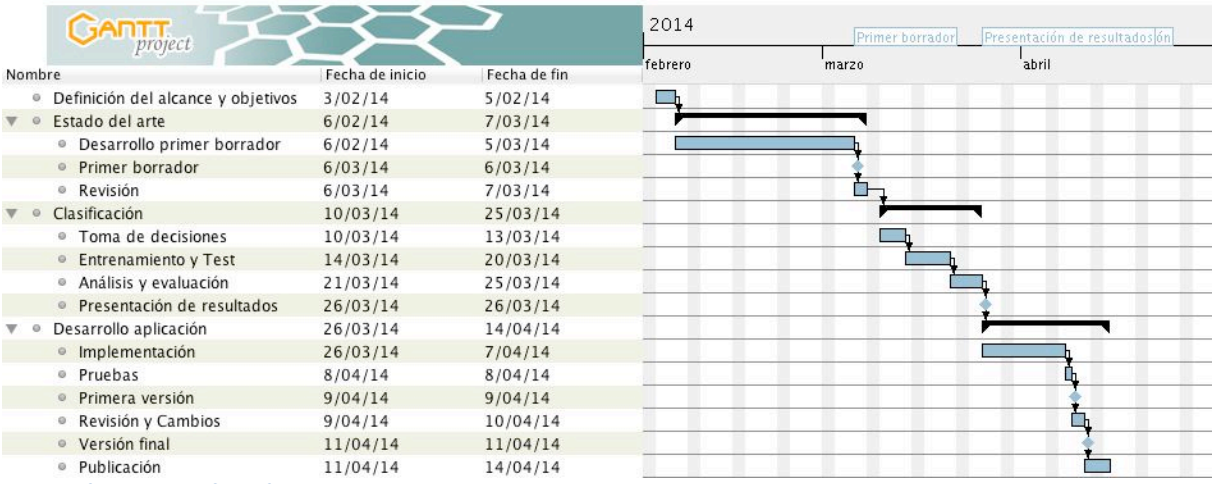
4.1.Planificación

La planificación del trabajo se ha realizado en varias fases:

- Definición de alcance y objetivos.
- Estado del arte.
- Clasificación.
- Desarrollo de aplicación.

Cada una de las fases esta compuesta por distintas tareas, revisión (realizadas por los tutores) y los cambios que se propusieron, toma de decisiones y por hitos que marcan los resultados intermedios y finales del trabajo.

La planificación esta comprendida entre el 3/02/14 al 14/04/14 como se muestra en la ilustración- 54.



Nombre	Fecha de inicio	Fecha de fin
Definición del alcance y objetivos	3/02/14	5/02/14
Estado del arte	6/02/14	7/03/14
Desarrollo primer borrador	6/02/14	5/03/14
Primer borrador	6/03/14	6/03/14
Revisión	6/03/14	7/03/14
Clasificación	10/03/14	25/03/14
Toma de decisiones	10/03/14	13/03/14
Entrenamiento y Test	14/03/14	20/03/14
Análisis y evaluación	21/03/14	25/03/14
Presentación de resultados	26/03/14	26/03/14
Desarrollo aplicación	26/03/14	14/04/14
Implementación	26/03/14	7/04/14
Pruebas	8/04/14	8/04/14
Primera versión	9/04/14	9/04/14
Revisión y Cambios	9/04/14	10/04/14
Versión final	11/04/14	11/04/14
Publicación	11/04/14	14/04/14

Ilustración 54 – Planificación

4.2.Presupuesto

Para la elaboración del presupuesto únicamente consideraremos la mano de obra, puesto que se necesitarán materiales, ni licencias de documentación o software adicionales.

El trabajo se considera en su totalidad como de “investigación y desarrollo” y realizada por un titulado en ingeniería informática.

Siguiendo la planificación y las consideraciones anteriores se han invertido un total de 306 horas invertidas. El coste estipulado por hora se considera de 60€.

A continuación se presenta una tabla con los resultados del presupuesto.

Concepto	Horas	Coste/hora	Coste total
Investigación y desarrollo	306	60€	18.360,00€
IVA (21%)			3.855,60€
TOTAL			22.215,60€

El coste total del trabajo propuesto asciende a veinte dos mil doscientos quince euros con sesenta céntimos.

5.Conclusiones

El objetivo de este trabajo ha sido realizar un estudio sobre la clasificación de documentos en castellano, basándose en el estándar IPTC. Para ello se ha utilizado un conjunto de ejemplos formado por documentos previamente etiquetados.

Con el algoritmo de clasificación seleccionado y los conjuntos de ejemplos de los que se disponía se ha realizado varias pruebas de clasificación (validación cruzada de 10 iteraciones).

Los resultados obtenidos (véase sección 2.3.5.) en el caso de categorizar en los 3 niveles que describe el estándar IPTC (sobre 1400 categorías posibles), muestran una clasificación poco satisfactoria al no alcanzar el 50% de precisión en ninguno de los casos propuestos y no cubrir la totalidad de las categorías propuestas.

Al realizar la clasificación en un primer nivel del estándar IPTC, es decir sobre 18 categorías posibles, los resultados se consideraron aceptables (superando el 50% y alcanzando el 65% y cubrir las 18 categorías propuestas).

Al estudiar los resultados de clasificación de artículos procedentes de las fuentes anteriormente mencionadas, reveló que los resultados de la clasificación varía dependiendo de las características de los artículos: longitud del artículo (unas veces muy largos y otras excesivamente cortos), vocabulario utilizado, ambigüedad del lenguaje, sarcasmo, metáforas, etc.

Con todo ello se concluye que la clasificación de documentos en lenguaje natural, y en particular los pertenecientes al lenguaje periodístico, no depende en su totalidad del modelo de clasificación utilizado, sino que se ve influenciado por las características del lenguaje empleado en la redacción del artículo.

Como prueba de concepto de la utilidad de la clasificación realizada, se ha decidido realizar una aplicación para la clasificación de artículos de noticias contenidos en feeds de noticias procedentes de distintos medios generalistas disponibles a través de internet.

La conclusión de este trabajo es que la clasificación de documentos en lenguaje natural es un campo que merece la pena ser explorado en profundidad, sirviendo como base para futuros proyectos, debido al gran número de aplicaciones en los que puede ser aplicado, muchas de ellas de manera comercial.

6.Bibliografía

- Sean Owen, Robin Anil, Ted Dunning, and Ellen Friedman. Mahout in action. Manning, 2011.
- Erik Hatcher, Otis Gospodnetic, Michael McCandless. Lucene in action, 2010
- http://docs.scala-lang.org/
- http://www.playframework.com/documentation/2.2.x/Home
- http://stackoverflow.com/tags/playframework
- https://developers.google.com/feed/v1/jsondevguide#request_format
- http://www.rss.nom.es/
- http://textalytics.com
- https://cwiki.apache.org/confluence/display/MAHOUT/Algorithms
- http://www.iptc.org/site/Home/
- http://www.layoutit.com/
- http://es.wikipedia.org/wiki/Wikipedia:Portada
- http://www.heroku.com
- https://devcenter.heroku.com/articles/getting-started-with-scala
- http://blog.classora.com/2012/10/10/describiendo-el-conocimiento-en-un-formato-estandar-para-la-web-semantica-rdf/
- http://blog.classora.com/2013/02/28/metadatos-definicion-aplicaciones-y-estandares/
- Información: http://www.rss.nom.es/lector-rss/
- http://tendenciasweb.about.com/od/nociones-basicas/a/Lectores-Rss-La-Forma-Mas-Sencilla-De-Estar-Pemanentemente-Informado.htm
- http://www.classora-technologies.com/es/