Universidad del País Vasco
Euskal Herriko Unibertsitatea

Máster Universitario en
Ingeniería Computacional y Sistemas Inteligentes

Konputazio Zientziak eta Adimen Artifiziala Saila –
Departamento de Ciencias de la Computación e Inteligencia Artificial

K I S A
I C S I

Tesis de Máster

ANÁLISIS Y CLASIFICACIÓN DE INFORMACIÓN MEDIÁTICA ELECTORAL UTILIZANDO MINERÍA DE TEXTO

Daríio Jiménez Paute

Tutor(a/es)

Aitor Soroa
Departamento de Ciencia de la Computación e Inteligencia Artificial
Facultad de Informática

Oier López
Departamento de Ciencia de la Computación e Inteligencia Artificial
Facultad de Informática

informatika fakultatea
facultad de informática

KZAA /CCIA

INDICE DE CONTENIDO

INDICE DE GRÁFICAS

INDICE DE TABLAS

1. INTRODUCCIÓN

La gestión del conocimiento cada vez es un recurso de mayor impacto e importancia estratégica para los organismos; así como para la generación, codificación, gestión y difusión de la información; aportando de esta manera al proceso de innovación.

Actualmente, el acelerado aumento en el volumen de datos disponibles de monitoreo de medios de comunicación es una realidad; y ha determinado una necesidad imperiosa para su procesamiento analítico, ya sea a través de aplicaciones de aprendizaje computacional, capaces de extraer información y conocimiento o de algoritmos de minería de texto que coadyuven al análisis y clasificación de los bancos de datos.

Esta información, más que base de datos, es a menudo un repositorio fundamental del conocimiento de organismos; pero no se encuentra estructurada. Cada día se dispone de más información y por tanto más patrones, con lo cual es necesario el uso de algoritmos eficientes con una convergencia a la solución óptima.

Por esta razón, la Minería de Texto es una tecnología emergente cuyo objeto es la búsqueda de conocimiento en grandes colecciones de documentos no estructurados, y se refiere al proceso de derivar información nueva de textos. La minería de texto consiste en, descubrir a partir de cantidades de textos grandes, el conocimiento que no está literalmente escrito en cualquiera de los documentos; esto incluye buscar tendencias, promedios, desviaciones, dependencias, entre otras. La minería de texto se apoya en varias técnicas como: clasificación de polaridad, procesamiento de lenguaje natural, extracción y recuperación de la información, aprendizaje automático, etc.

En resumen, la minería de texto puede ayudar a que la información implícita se muestre en documentos más claros y que ahorre tiempo y dinero a los

organismos. Motivo por el cual la importancia de realizar un análisis y clasificación de información mediática electoral utilizando la minería de texto.

1.1 Planteamiento del problema

La clasificación de polaridad para textos dado a nivel de documento, oración, o rasgo/característica consiste en, la opinión expresada de un organismo en positiva, negativa, o neutral.

Así mismo; en la actualidad el formidable aumento de la información a través de medios de comunicación, ha determinado que el análisis manual de los expertos en las diferentes temáticas, se vea afectado para su procesamiento inmediato; ya que implica un esfuerzo adicional para alcanzar el objetivo de información oportuna para la toma de decisiones.

Las temáticas electorales no son una excepción en este aspecto, la ejecución dinámica del sistema democrático del país, a través de los diferentes mecanismos de democracia directa como lo son: iniciativa popular, consultas populares, y revocatorias de mandato; ha abocado a que se multipliquen las actividades electorales, y por lo tanto su difusión y recogimiento a través de los diferentes medios de comunicación (prensa, radio, tv, digitales), aumentado en un volumen considerable.

Con lo cual, el proceso de monitoreo de medios para las diferentes temáticas electorales, recopila las noticias con ciertos criterios político-electorales, y de carácter informativo, donde se conoce cada día desde primera hora las menciones que los medios de comunicación realizan sobre organismos electorales, sus miembros destacados, su competencia o su ámbito de acción.

La utilidad práctica de esta aproximación de clasificación, será avalada por los resultados experimentales que muestren una mejora en el enfoque tradicional, en la cual los expertos en el dominio de textos y equipos completos de análisis definen manualmente las reglas de clasificación.

1.2 Objetivo del estudio

El objetivo del presente estudio, es poder crear un modelo de predicción a partir de datos de las noticias electorales reales; es decir, con el aprendizaje supervisado y con el conjunto de datos, crear un modelo con técnicas de aprendizaje automático que clasifique la polaridad de las noticias electorales, en positivas, negativas o neutrales; optimizando de esta manera los tiempos de procesamiento y la oportunidad de la información para la toma de decisiones en el organismo electoral.

Además, con este estudio se pretende abordar el procesamiento del conjunto de datos desbalanceado, con la efectividad de los clasificadores a través de las diferentes métricas de evaluación. Así mismo, se estima comparar de manera estadística los resultados del conjunto de datos de entrenamiento y test, y comprobar si realmente no habido un sobreajuste en el aprendizaje del modelo para determinar el mejor.

1.3 Distribución de la tesis

Para el presente documento, se inicia en la Sección 2 describiendo el conjunto de datos de las noticias electorales que han sido utilizados como parte del repositorio fundamental del conocimiento mediático de los organismos en un período determinado. La Sección 3, muestra el marco experimental con la definición de conjuntos de datos y su respectiva

metodología. En la Sección 4, se describe la exploración de filtros teniendo como base un clasificador.

Se continúa el trabajo con la exploración de clasificadores en la Sección 5, para luego plasmar los resultados y análisis de estos experimentos en la Sección 6, las conclusiones del objeto de estudio se describirán en la Sección 7, y finalmente se puntualizará algunas recomendaciones y trabajo a futuro en la Sección 8.

2. DESCRIPCIÓN DEL CONJUNTO DE DATOS

Para el presente estudio se ha tomado como base el conjunto de datos obtenido del monitoreo de noticias de los medios electrónicos, impresos, televisivos y radiales, en el período Marzo 2015 a Marzo2016; del conjunto de datos conformado por varios campos de acuerdo a la siguiente estructura de datos:

Gráfica N° 1 Estructura datos

Descripción de los campos del gráfico:

- Año: año de publicación de la noticia.
- Mes: mes de publicación de la noticia.
- Día: día de publicación de la noticia.
- Tipo Medio: tipo de medio de comunicación (radio, tv, prensa escrita, web) donde se publicó la noticia.
- Medio: nombre del medio de comunicación donde se publicó la noticia.
- Programa: nombre del espacio televisivo, radial, escrito o web en el cual se ha difundido la noticia
- Titular: titular que lleva la noticia
- Resumen: extracto de la noticia.
- Color: color de la publicación (orientada a prensa escrita)
- Tipo: tipo de noticia (positiva, negativa, neutral).
- Vocero: autoridades y/o representantes institucionales que forman parte de la difusión de la noticia

De los cuales, se ha seleccionado como campo a ser analizado y sujeto a la experimentación el **tipo** de **noticia**; por la importancia de su procesamiento y resultados en la toma de decisiones, y cuya clasificación luego de haber recopilado la información se realiza de manera manual por el equipo de expertos y de análisis.

Luego del preprocesamiento se ha determinado utilizar la base de datos de noticias con dos campos importantes (resumen, tipo); en el **resumen** se ha unido el titular de la noticia, ya que este condensa el tema principal de la noticia; así mismo, debido a la relevancia de la clasificación el **tipo de noticia** que muestra el posicionamiento y percepción mediática en materia electoral; con lo cual tendríamos la siguiente estructura de base de datos:

Gráfica N° 2 Estructura definida datos

Resumen	Tipo

Los dos campos contienen la siguiente información:

Resumen: atributo que recoge el titular y el resumen de la noticia publicada en un medio de comunicación, de acuerdo al análisis y extracción de la información; con un promedio de longitud de 221 palabras, de acuerdo al tipo de noticia que se muestra en la Tabla N°1.

Tabla N° 1 Promedio de palabras por clase

	Neutrales	Positivas	Negativas
Promedio palabras	271	205	189

De igual forma, tenemos la frecuencia de palabras de acuerdo al tipo de noticia, que se detalla en la siguiente tabla:

Tabla N° 2 Frecuencia de términos en clases

Neutrales fr=2500	Positivas fr = 500	Negativas fr =50
alianza	consejo	cne
comisión	electoral	consejo
consejo	nacional	control
consulta	manga	electoral
ecuador	consulta	nacional
electoral	cne	participación
enmiendas	popular	ciudadana
gobierno	firmas	gobierno
ley	organizaciones	proceso
movimiento	políticas	presidente
nacional	presidente	consulta
organizaciones	proceso	observación
país		
presidente		
proyecto		

Como se puede observar en la Tabla N°2 , la frecuencia de los términos no nos aporta significativamente para el pre procesamiento; sin embargo se puede evidenciar términos comunes en los tipos de noticias, como por ejemplo: consulta, presidente, consejo, nacional y electoral.

Tipo: atributo que define el tipo de noticia de acuerdo a su clasificación neutral (0), positiva (1) y negativa (2).

Positivas.- Una noticia se considera positiva, si contiene información y datos sobre actividades o hechos relativos al organismo electoral, con opiniones positivas, resaltando o valorando positivamente las acciones o posiciones, con interpretaciones de apoyo, respaldo de forma positiva hacia el organismo electoral, desde los sujetos noticiosos o desde el medio de comunicación.

> **Ejm:** *'Próximo año se abrirá Museo de la Democracia La creación de la Biblioteca y Museo de la Democracia es el objetivo a cumplir para el 2016 por parte de la Delegación Provincial Electoral de Tungurahua. La directora indicó que el museo tendrá el carácter histórico y político electoral. Esta será la primera ocasión en que la provincia disponga de una memoria única que pueda ser dada a conocer respecto a cómo se dieron las primeras elecciones\ así como el conocer las primeras actas\ papeletas\ organizaciones y fuerzas políticas que han sido parte de la historia.';1*

Negativas.- Una noticia se considera en negativa, si contiene información errada, datos incorrectos, críticas en contra del organismo electoral, opiniones negativas con respecto al organismo electoral, sus acciones o posiciones, interpretaciones equivocadas o negativas de parte de los sujetos noticiosos o desde el medio de comunicación. Críticas desde sectores opositores a los organismosdel Estado. Inclusive se categoriza así cuando parcialmente dentro de la misma noticia se emitan también, como parte de ella, comentarios positivos.

> **Ejm:** *'Denuncian supuesto fraude electoral en consulta de la Manga del Cura Los representantes de los recintos Santa Teresa\ Paraíso de la 14 y La Caoba pertenecientes a la Manga del Cura\ rechazan los resultados de la consulta popular realizada el pasado 27 de septiembre en ese sector.';2*

Neutrales.- Una noticia se clasifica en neutral, si contiene información o datos sobre actividades del organismo electoral o relacionadas con este, sin opiniones parcializadas con respecto al organismo electoral, sin

interpretaciones parcializadas de parte de los sujetos noticiosos o desde el medio de comunicación. Hechos o noticias relativos al organismo electoral desde otras funciones del Estado y que no tengan connotación positiva o negativa.

> **Ejm:** *'Un recurso sin precedente es la revocatoria. Ante el anuncio de la petición de revocatoria de mandato a los asambleístas de Alianza PAIS en todo el país\ se analizó el procedimiento con la coordinación provincial del organismo electoral\ institución que está a cargo de este tipo de procedimientos. Ante esto la directora provincial del organismo electoral, describió que este es un proceso extenso en el que se deben cumplir varios requisitos. El primero es que el representante electo debe tener al menos 1 año en función de su cargo y debe faltar 1 antes que termine su periodo. En el caso de los asambleístas por Chimborazo se encuentran en su segundo año de mandato\ por cual se podría considerar esta posibilidad.';0*

Finalmente el conjunto de datos quedaría conformado por 34650 instancias del período Marzo 2015 a Marzo 2016, que corresponden a 31150 instancias de tipo neutral que representa el 89.9% del conjunto de datos, 3180 instancias de tipo positiva que representa el 9.2%, y 320 instancias de tipo negativa que representa el 0.9%, de acuerdo a la siguiente tabla:

Tabla N° 3 Conjunto de datos

Tipo	Instancias	Porcentaje
Neutral (0)	31150	89,9%
Positiva (1)	3180	9,2%
Negativa (2)	320	0,9%
	34650	100%

Como se muestra en la Tabla N° 3, el conjunto de datos principal está desbalanceado en dos de sus clases, y mucho más en la tercera clase negativa (2), que es una de las más importantes a la hora del análisis de la información.

3. MARCO EXPERIMENTAL

En el presente marco experimental, se estudiará diferentes métodos supervisados para el análisis de la polaridad de documentos, partiendo de la definición del conjunto de datos, posteriormente definición de las métricas de evaluación; para luego realizar la exploración de filtros y clasificadores, de acuerdo a las métricas de evaluación seleccionadas, terminando con los resultados y el análisis de los valores obtenidos en los experimentos.

Esta metodología definida para los siguientes experimentos, se describe a continuación:

3.1 Definición de conjuntos de datos

Para el análisis del objeto de estudio se han definido tres conjuntos de datos: entrenamiento, desarrollo y test, manteniendo la distribución de clases en cada conjunto, con los cuales se experimentará el modelo. A continuación, tenemos la conformación de los conjuntos de datos y su integración de clases:

Tabla N° 4 Definición conjuntos de datos

		Entrenamiento	Desarrollo	Test
Tipo		**40%**	**30%**	**30%**
Neutral	31150	12460	9345	9345
Positiva	3180	1272	954	954
Negativa	320	128	96	96
Total	**34650**	**13860**	**10395**	**10395**

Como se observa en la Tabla N° 4, el conjunto de datos de entrenamiento, nos servirá para entrenar el modelo supervisado en cada uno de los

experimentos; el conjunto de datos de desarrollo nos permitirá probar y evaluar el modelo construido con los datos de entrenamiento, y finalmente el conjunto de datos test, permitirá probar y evaluar el modelo con las mejores configuraciones obtenidas en los experimentos con el conjunto de datos de desarrollo.

3.1.1 Definición de la línea base conjunto de datos entrenamiento

Al observar la Tabla N°4 se aprecia perfectamente el desbalance del conjunto de datos, y que al realizar experimentos con los algoritmos de clasificación, es necesario tener alguna referencia para interpretar la significancia de los resultados; por lo que se definió una línea base (baseline) para objeto de comparación e interpretación de resultados de la clasificación automática.

Para la definición del baseline se consideran dos sistemas, en el primero, denominado sistema “BL1”, se etiquetan todas las instancias clasificadas (tabla N° 5), es decir, para el caso de neutrales, en las 13860 instancias indicará que sí son neutrales, es decir sabemos que acertará en 12460 de ellas, por lo que su efectividad será del 89.9%. En el segundo, denominado sistema “BL2”, se etiquetan todas las instancias sin la clasificación neutral, es decir y a manera de ejemplo, para el caso de la clase neutrales, viendo la tabla N°. 5 sabemos que acertará en 1400 de ellas, por lo que su efectividad será del 10.1%.

Tabla N° 5 Línea base (baseline)

Tipo	Instancias entrenamiento	BL1	BL2
Neutrales	12460	89,9%	10,1%
Positivas	1272	9,2%	90,8%
Negativas	128	0,9%	99,1%
Total	**13860**	100,00%	200,00%
	Promedio	**33,3%**	66,7%

Para el presente estudio se tomará como línea base (baseline) el sistema "BL1" con un promedio del 33.3%, ya que se estima aplicar la métrica de evaluación macro-f1, en la cual las tres clases tendrían la misma importancia al implementar los experimentos de clasificación.

3.2 Métricas de evaluación

Son medidas muy importantes que permiten la evaluación del rendimiento y efectividad de los diferentes clasificadores, por lo que en el presente estudio se han definido las siguientes métricas que se utilizarán para evaluar los filtros y clasificadores, mismos que determinarán el mejor clasificador:

Estas métricas se basan en el número de verdaderos positivos (VP), el número de falsos positivos (FP), el número de verdaderos negativos (VN), y el número de falsos negativos (FN) en los datos; como se muestra en la siguiente tabla:

Tabla N° 6 Métricas

	Pertenece	No pertenece
Seleccionado	VP	FP
No seleccionado	FN	VN

Precision es denominada como la probabilidad de que un documento cualquiera sea clasificado bajo su categoría:

$$P = \frac{VP}{VP+FP}$$

Recall es una medida de capacidad de un modelo de predicción para seleccionar instancias de una clase determinada de un conjunto de datos:

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

F1 esta es la media armónica de la precision y el recall:

$$F1 = \frac{2PR}{P+R}$$

Para encontrar una métrica que represente la efectividad de un clasificador, teniendo en cuenta tanto la precision, recall y f1, de modo que nos permita tener una idea más general de cuan buenos son los resultados, utilizaremos lo que se conoce como micro-f1 y macro-f1.

Macro-f1: Se calcula el rendimiento (precision, recall, f1) de cada clase por separado y luego se obtiene el promedio.

Micro-f1: Se suman los falsos positivos de todas las clases, los falsos negativos, etc. Y con eso se calcula precision y recall. Aquí dominan las clases más frecuentes. (Camelo, Bernal, & Barreras, 2015).

Para el presente estudio la métrica que se ha considerado para la evaluación integral de los experimentos, ha sido el macro-f1, ya que esta métrica toma en cuenta todas las clases y le da el mismo peso a las tres clases que forman parte de nuestro conjunto de datos. Además, al tener los conjuntos de datos desbalanceados, y al ser de vital importancia para el organismo electoral conocer cuáles noticias son clasificadas con la clase 2 (noticias negativas), el macro-f1 brindará un valor que permita evaluar integralmente al clasificador.

3.3 Herramienta de trabajo

En el presente estudio se ha determinado utilizar en el proceso de experimentación la herramienta automatizada WEKA (Waikato Environment for Knowledge), que es un conjunto de algoritmos de aprendizaje automático para tareas de minería de datos. Weka contiene herramientas para el procesamiento previo de datos, clasificación, regresión, clustering, reglas de asociación, y la visualización. Trabaja para su procesamiento con un formato de datos **.arff**, acrónimo de Atribute Relation File Formato. (Ian, Eibe, & Mark, 2011).

Weka, implementa algoritmos que pueden aplicarse al realizar el preprocesamiento de datos, para transformarlos en un esquema de aprendizaje a fin de que los resultados puedan ser analizados de manera sencilla; es decir, permite aplicar métodos de aprendizaje a conjuntos de datos y analizar los resultados para extraer información.

3.4 Exploración de filtros y clasificadores (conjunto de datos desarrollo)

En esta fase de exploración se utilizarán los conjuntos de datos de **entrenamiento** y **desarrollo** para la extracción de atributos a partir de los

documentos. Estos atributos primeramente deberán ser convertidos en números, ya que los conjuntos de datos son textuales; para luego realizar la aplicación de los filtros como: la identificación de las palabras con su separación por tokens, la eliminación de las palabras vacías que tienen una frecuencia elevada pero que no aporta información relevante del documento la frecuencia de los términos en los documentos, y la normalización de los mismos.

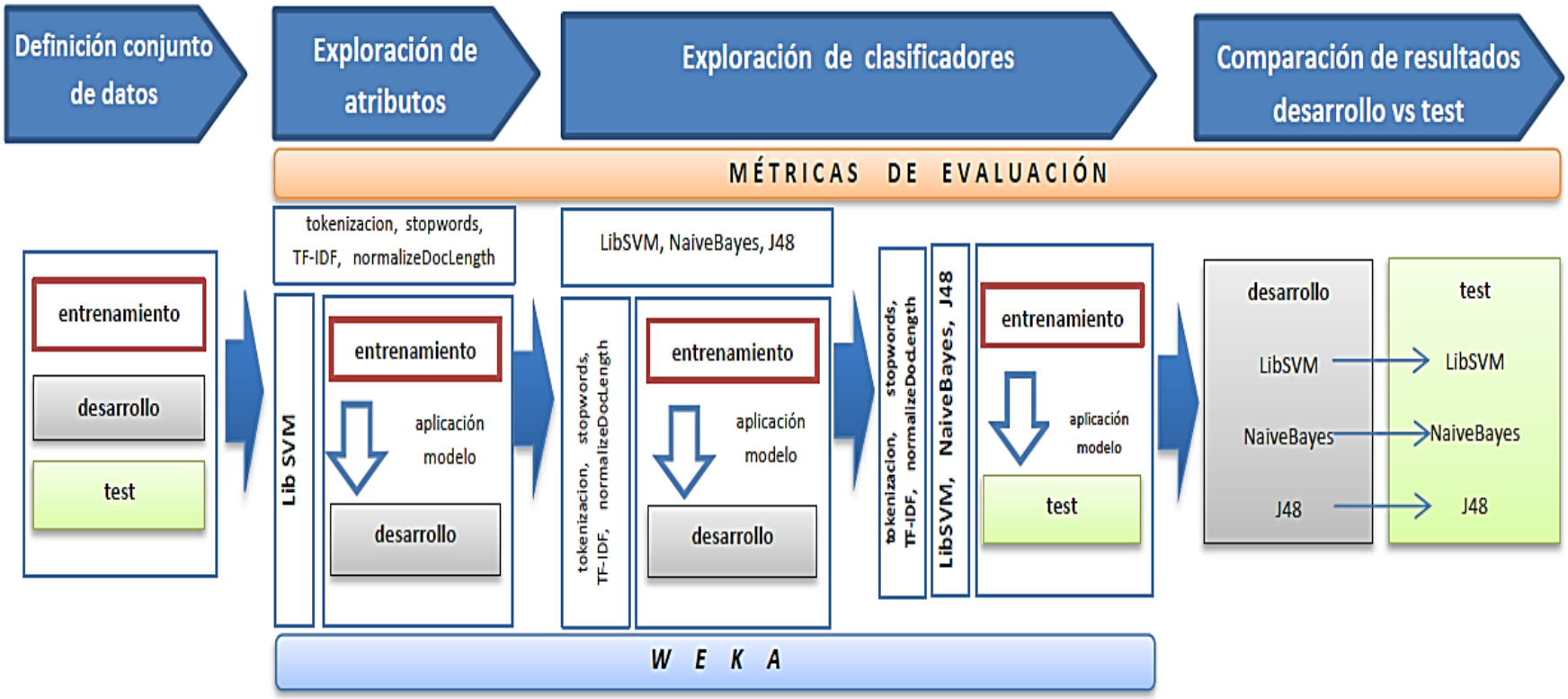
Así mismo, durante este proceso se utilizará un clasificador por defecto para obtener los valores de configuración de los filtros- Los mejores resultados de estos experimentos serán utilizados en la siguiente fase, que consiste en la exploración de los tres clasificadores seleccionados en el presente objeto de estudio con sus parámetros por defecto, y sus parámetros críticos optimizados.

3.5 Exploración de clasificadores (conjunto de datos test)

Una vez obtenidas las mejores configuraciones de los filtros y clasificadores de la sección anterior; con el conjunto de datos de desarrollo, procederemos a probar y evaluar el modelo con el conjunto de datos test.Posteriormente, realizaremos la comparación de los valores obtenidos con el conjunto de datos de desarrollo y conjunto de datos test, para analizar los resultados e identificar si no existe algún sobreajuste del modelo.

A continuación en la gráfica N° 3, se ilustra la metodología del marco experimental, a través de un flujograma de información:

Gráfica N° 3 Flujo metodología



4. EXTRACCIÓN DE ATRIBUTOS

Es posible crear modelos de clasificación de documentos de texto en categorías previamente analizadas. Los documentos normalmente necesitan ser convertidos en “vectores de texto” antes de aplicar las técnicas del aprendizaje automático. Para esto la manera más sencilla de representar el texto es como bolsa de palabras o palabra vector. Un filtro puede realizar el proceso de convertir el atributo string a un conjunto de atributos que representara la ocurrencia de las palabras del texto completo. El documento se representa como una cadena de texto que se constituye como un solo atributo de tipo string.

Un filtro fundamental para el análisis de texto, se conoce como StringToWordVector. En este filtro se pueden configurar todas las técnicas del procesamiento lingüístico del lenguaje natural a los atributos, ofrece abundantes opciones del procesamiento del lenguaje natural; sin embargo para los experimentos a desarrollar, aplicaremos los siguientes: tokens personalizados, uso de listas de palabras vacías, cálculo TF-IDF frecuencia de ocurrencia del término en la colección de documentos, y normalización.

La estructura del conjunto de datos, previo al procesamiento a través de los filtros de esta herramienta es la siguiente:

@relation 'NotiTrain'

@attribute Resumen string

@attribute Tipo {0,1,2}

@data

'César Montúfar: ?La reunión en Cuenca marca el momento preelectoral? Su punto de vista. Hay tres puntos sobre los que debe actuar la oposición: luchar contra las enmiendas\ consolidar la unidad en torno

a candidaturas únicas para el 2017 y plantear un modelo político que reemplace el modelo económico y político del correísmo. ',0

'Avanza formará una escuela de capacitación política La directiva de Guayas del partido Avanza anunció la creación de una escuela de capacitación política para fortalecer a la agrupación\ con miras a las elecciones de 2017. El presidente nacional de la organización política\ Ramiro González\ visitó Guayaquil para conocer pormenores de este proyecto en la ciudad.',0

……..

……..

……..

A continuación tenemos los filtros explorados para el presente estudio:

4.1 Tokenización

Elije unidad de medida para separar el atributo de texto completo. Para identificar las distintas palabras (o tokens) es necesario primero definir los delimitadores de los tokens, los que generalmente corresponden a los signos de puntuación y otros caracteres distintos a las letras del alfabeto.

Luego los delimitadores de tokens se separan de las palabras y son reemplazados por un espacio blanco simple. De esta forma cada palabra queda separada por un espacio blanco simple y facilita la tokenización.

4.2 Stopwords

Las palabras vacías o stopwords son los términos que se han generalizado y aparece con más frecuencia, no proporcionan información sobre un texto. Esta opción determina si una sub cadena en el texto es una palabra vacía. Las palabras vacías provienen de una lista predefinida. Esta opción convierte todas

las palabras del texto minúsculas antes de la eliminación. Stopwords es pertinente para eliminar palabras sin sentido dentro del texto y eliminar de las palabras frecuentes y útiles de árboles de decisión, como este por ejemplo “de”.

Las palabras vacías de Weka por defecto se basan en las listas de Rainbow (Rainbow, 1998)/. El formato de estas listas es una palabra por línea donde los comentarios serán cada línea que comienzan con ‘#’ para ser omitidos.

4.3 Tf-idf

(Term frequency – Inverse document frequency), frecuencia de término – frecuencia inversa de documento (o sea, la frecuencia de ocurrencia del término en la colección de documentos), es una medida numérica que expresa cuán relevante es una palabra para un documento en una colección. Esta medida se utiliza a menudo como un factor de ponderación en la recuperación de información y la minería de texto. El valor tf-idf aumenta proporcionalmente al número de veces que una palabra aparece en el documento, pero es compensada por la frecuencia de la palabra en la colección de documentos, lo que permite manejar el hecho de que algunas palabras son generalmente más comunes que otras.

4.4 NormalizeDocLength

Normalizar el tamaño del documento para establecer si las frecuencias de las palabras en una instancia deben ser normalizados. La normalización es calculada como:

ValorActual * longitudDelDocumentoPromedio /
realLongitudDelDocumento.

5. EXPLORACIÓN DE CLASIFICADORES

En esta etapa, los documentos procesados por los filtros de la etapa anterior, se utilizarán para construir el modelo de acuerdo a la exploración de clasificadores que asignará categorías a los documentos. Evaluaremos tres clasificadores como lo son LibSVM, NaiveBayes y J48.

Es importe considerar que la limitación del aprendizaje supervisado, es la disponibilidad de una cantidad importante de ejemplos de entrenamiento ya clasificados; y esta categorización o etiquetado es un proceso manual y laborioso, de acuerdo a la problemática planteada.

Sin embargo, en el presente estudio de acuerdo a lo descrito en la sección de la descripción del conjunto, contamos con una importante cantidad de registros para los experimentos con los clasificadores.

5.1 LibSVM

Es una implementación para la máquina de vectores soporte (SVM), cuyo objetivo es aplicar fácilmente el clasificador SVM en las aplicaciones, teniendo una gran popularidad en cuanto al aprendizaje automático.

El uso típico de LIBSVM en su tarea de clasificador contiene dos pasos: primero, entrenamiento de un conjunto de datos para obtener un modelo y segundo, usar el modelo obtenido para predecir información de un conjunto de datos test, tanto para SVC(clasificación) y SVR(Regresión), LIBSVM aparte de entregar la precisión del sistema, puede también tener como salida: probability estimates, que puede interpretarse como la probabilidad estimada de acierto (verdadero positivo) o de error (falso positivo).

Para iniciar la experimentación debemos considerar la configuración de parámetros por defecto de este clasificador:

Seed.- La semilla de números aleatorios para ser utilizado. *(por defecto 1)*

Pérdida.- El épsilon de la función de pérdida de épsilon-SVR. *(por defecto 0.1)*

modelFile - El archivo para guardar el modelo libsvm-interno para; ningún modelo se guarda si apunta a un directorio.

kernelType.- El tipo de núcleo para usar *(por defecto radial)*

numDecimalPlaces.- El número de cifras decimales que se utilizarán para la salida de los números en el modelo. *(por defecto 2)*

batchSize.- El número preferido de instancias de proceso si se realiza la predicción por lotes. Más o menos casos se pueden proporcionar, pero esto le da la oportunidad de implementaciones especificar un tamaño de lote preferido. *(por defecto 100)*

cacheSize.- El tamaño de la caché en MB. *(por defecto 40.0)*

degree.- El grado de kernel. *(por defecto 3)*

gamma - La gamma de usar, si es 0 entonces se utiliza 1 / max_index. *(por defecto 0.0)*

Shrinking.- Si hay que usar la heurística de la contracción. *(por defecto True)*

eps.- La tolerancia del criterio de terminación. *(por defecto 0.001)*

cost.- El parámetro de coste C para C-SVC, épsilon-SVR y nu-SVR. *(por defecto 1.0)*

SVMType - El tipo de SVM para su uso. *(por defecto C_SVC)*

weights.- Los pesos a utilizar para las clases (lista en blanco separado, por ejemplo, "1 1 1" para un problema de clase 3), si está vacío 1 se utiliza por defecto.

normalize - Ya sea para normalizar los datos.

coef0 - El coeficiente de usar. *(por defecto 0.0)*

nu - El valor de las nu para nu-SVC, SVM una sola clase y nu-SVR. *(por defecto 0.5)*

Para el análisis y optimización de los parámetros críticos de este clasificador, se utilizó la herramienta CVParameters, que es un “meta" clasificador que busca automáticamente los “mejores" valores de los parámetros mediante la optimización cross-validation la exactitud de los datos de entrenamiento. (Optimizing parameters)

Cada ajuste se evaluó utilizando 3 y 4 veces respectivamente la validación cruzada. Para cada uno de ellos, tenemos que dar (a) una cadena que le asigna el nombre utilizando su código de letras, (b) una serie de valores numéricos para evaluar, y (c) el número de pasos para tratar en este rango.

Los parámetros críticos que se han tomado para el análisis y experimentación han sido los siguientes: **gamma (G)** y **cost (C).** Configurados para su optimización en CVParameteres tenemos:

G 0.01 0.1 3.0

Cross-validation parámetro: '-C' osciló desde 0.01 a 0.1 con 3.0 pasos

Opciones clasificador: **-G 0.01 -S 0 -K 2 -D 3 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -model C:\Weka-3-8 -seed 1**

C 2.0 8.0 4.0

Cross-validation parámetro: '-M' osciló desde 2.0 a 8.0 con 4.0 pasos

Opciones clasificador **-C 8 -S 0 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -E 0.001 -P 0.1 -model C:\Weka-3-8 -seed 1**

Los valores optimizados para los parámetros seleccionados, luego de estos resultados automáticos en la herramienta Weka son:

G = 0.01 y **C** = 8

Cabe recalcar que el costo de procesamiento fue muy elevado para la optimización de los parámetros, por lo cual no se pudo incrementar más pasos para estos experimentos.

5.2 Naive Bayes

El clasificador Naive Bayes es un clasificador probabilístico basado en el supuesto de que el valor de un atributo en una clase dada es independiente de los valores de los otros atributos, lo que permite hacer simplificaciones importantes en los cálculos involucrados. Este clasificador es uno de los métodos más simples y fáciles de implementar; adicional, se basa en el teorema de Bayes y en la premisa de independencia de los atributos dada una clase. Esta premisa es conocida como “Naïve Assumption" y se le llama “naïve" (o ingenua) considerando que en la práctica los atributos raramente son independientes.

Para iniciar la experimentación debemos considerar la configuración de parámetros por defecto de este clasificador:

> batchSize - El número preferido de instancias de proceso si se realiza la predicción por lotes. Más o menos casos se pueden proporcionar, pero esto le da la oportunidad de implementaciones especificar un tamaño de lote preferido. *(por defecto 100)*
>
> numDecimalPlaces - El número de cifras decimales que se utilizarán para la salida de los números en el modelo. *(por defecto 2)*
>
> useKernelEstimator - Utilice un estimador kernel para atributos numéricos en lugar de una distribución normal. *(por defecto FALSE)*
>
> useSupervisedDiscretization - Uso supervisado discretización para convertir los atributos numéricos de los nominales. *(por defecto FALSE)*

Los parámetros que se han tomado para el análisis y experimentación han sido los siguientes: useKernelEstimator **(K)** y useSupervisedDiscretization **(D)**

5.3 J48

Es la implementación del algoritmo C4.5, el cual mide la cantidad de información contenida en un conjunto de datos y la agrupa por importancia. Se da la idea de la importancia de un atributo en un conjunto de datos. J48 imprime recursivamente la estructura del árbol con variables de tipo string mediante el acceso a la información de cada atributo almacenada en los nodos.

Para iniciar la experimentación debemos considerar la configuración de parámetros por defecto de este clasificador:

seed.- La semilla utilizada para la aleatorización de los datos cuando se utiliza la poda de errores reducida. *(por defecto 1)*

confidenceFactor.- El factor de confianza utilizado para la poda (valores menores incurren en más de poda). *(por defecto 0.25)*

numFolds.- Determina la cantidad de datos que se utilizan para la poda de errores reducida. Uno veces se utiliza para la poda, el resto para el crecimiento del árbol. *(por defecto 3)*

numDecimalPlaces.- El número de cifras decimales que se utilizarán para la salida de los números en el modelo. *(por defecto 2)*

batchSize.- El número preferido de instancias de proceso si se realiza la predicción por lotes. Más o menos casos se pueden proporcionar, pero esto le da la oportunidad de implementaciones especificar un tamaño de lote preferido. *(por defecto 100)*

subtreeRaising.- si se debe considerar la operación subárbol elevación al podar. *(por defecto True)*

minNumObj.- El número mínimo de instancias por hoja. *(por defecto 2)*

useMDLcorrection.- si la corrección MDL se utiliza cuando la búsqueda se divide en atributos numéricos. *(por defecto True)*

collapseTree.- Si se retiran las piezas que no reducen la formación de error. *(por defecto True)*

Para el análisis y optimización de los parámetros críticos de este clasificador, se utilizó la herramienta CVParameters. .Cada ajuste se evaluó utilizando cinco y dos veces respectivamente la validación cruzada.

Los parámetros críticos que se han tomado para el análisis y experimentación han sido los siguientes: confidenceFactor **(C)** y minNumObj **(M).** Configurados para su optimización en CVParameteres tenemos:

C 0.1 0.5 5.0

Cross-validation parámetro: '-C' osciló desde 0.1 a 0.5 con 5.0 pasos

Opciones clasificador: **-C 0.1 -M 2**

M 1.0 5.0 2.0

Cross-validation parámetro: '-M' osciló desde 1.0 a 5.0 con 2.0 pasos

Opciones clasificador **-M 5 -C 0.25**

Los valores optimizados para los parámetros seleccionados, luego de estos resultados automáticos en la herramienta Weka son:

C = 0.1 y **M** = 5

De igual manera, en este clasificar el costo de procesamiento fue elevado en la optimización de los parámetros críticos seleccionados.

6. RESULTADOS Y ANÁLISIS

En esta sección se presentan los resultados y análisis obtenidos a partir de la experimentación, en la exploración de los filtros y clasificadores de los conjuntos de datos descriptos en la sección 4 y 5.

Para medir la efectividad de los clasificadores en las experiencias nos basamos en la medida del macro-f1, determinada para el presente estudio, en virtud del conjunto de datos desbalanceado.

Todas las experiencias presentadas en esta sección fueron ejecutadas sobre los conjuntos de datos de entrenamiento, de desarrollo y test respectivamente. Así mismo, se realiza un análisis de los resultados del conjunto de datos de desarrollo versus el conjunto de datos test.

6.1 Resultados exploración de filtros

Para la experimentación y exploración de los filtros, se ha tomado como clasificador base el *LibSVM (Librería de las Máquinas de Vectores Soporte)* con sus parámetros por defecto, a continuación las tablas muestran las métricas precision, recall, f1, micro-f1 y macro-f1: .

Tabla N° 7 Resultados filtros por defecto

Preproceso Desarrollo	Neutral	positiva	negativa		instancias	Precision	Recall	f1
(por defecto)	9343	2	0	neutral	9345	90,2%	100,0%	94,9%
	914	40	0	positiva	954	95,2%	4,2%	8,0%
	96	0	0	negativa	96	0,0%	0,0%	0,0%

Micro-f1	**86,0%**
Macro-f1	34,3%

Tabla N° 8 Resultados filtros por defecto + tokenización

Preproceso Desarrollo	neutral	positiva	negativa		instancias	Precision	Recall	f1
por defecto + tokenización	9280	65	0	neutral	9345	92,0%	99,3%	95,5%
	715	239	0	positiva	954	78,4%	25,1%	38,0%
	95	1	0	negativa	96	0,0%	0,0%	0,0%

Micro-f1	**89,3%**
Macro-f1	44,5%

Tabla N° 9 Resultados filtros por defecto + tokenización + stopwords

Preproceso Desarrollo	neutral	positiva	negativa		instancias	Precision	Recall	f1
por defecto + tokenización + stopwords	9280	65	0	neutral	9345	92,0%	99,3%	95,5%
	715	239	0	positiva	954	78,4%	25,1%	38,0%
	95	1	0	negativa	96	0,0%	0,0%	0,0%

Micro-f1	**89,3%**
Macro-f1	44,5%

Tabla N° 10 Resultados filtros por defecto + tokenización + stopwords + TF IDF

Preproceso Desarrollo	neutral	positiva	negativa		instancias	Precision	Recall	f1
por defecto + tokenización + stopwords +TF IDF	9262	83	0	neutral	9345	93,2%	99,1%	96,0%
	589	365	0	positiva	954	80,6%	38,3%	51,9%
	91	5	0	negativa	96	0,0%	0,0%	0,0%

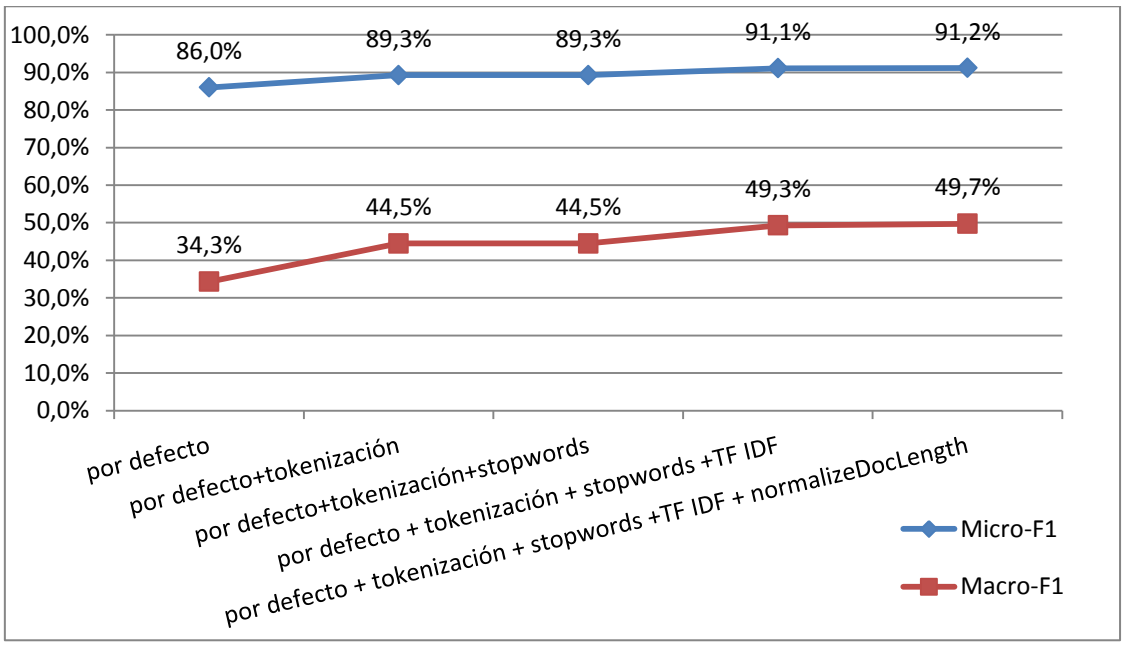
Micro-f1	**91,1%**
Macro-f1	49,3%

Tabla N° 11 Resultados filtros por defecto + tokenización + stopwords + TF IDF + NormalizeDocLength

Preproceso Desarrollo	neutral	positiva	negativa		instancias	Precision	Recall	f1
por defecto + tokenización + stopwords +TF IDF + normalizeDoc Length	9258	87	0	neutral	9345	93,2%	99,1%	96,1%
	577	377	0	positiva	954	81,1%	39,5%	53,1%
	95	1	0	negativa	96	0,0%	0,0%	0,0%

Micro-f1	**91,2%**
Macro-f1	49,7%

Gráfica N° 4: Efectividad filtros



Con las configuraciones por defecto de este clasificador, en un lapso de 29.55 segundos de procesamiento para la construcción del modelo en los datos de entrenamiento y 20.53 segundos para probar el modelo en los datos de desarrollo, se obtiene una precision del 89.9%, y recall del 90.3%, con lo cual la efectividad del clasificador con esta configuración es del 86% de acuerdo al micro-f1, mientras que el 34.3% de rendimiento de acuerdo al macro-f1.

Con la configuración adicional de tokenización en este método, se obtiene un 89,9% de precision, similar a la anterior configuración, y un recall del 91.6%; así mismo, se evidencia una mejora considerable del macro-f1 al 44.5%, acercando una mejor efectividad en la clasificación de todas las clases, mientras que el micro-f1 tiene una mejora mínima con relación a la anterior configuración.

Adicionando a la configuración anterior, el atributo stopwords en este método mantiene los valores de precision en 89.9% y un 91.6% de recall, con lo cual los resultados obtenidos del micro-f1 también mantiene su porcentaje de 86.3% y el macro-f1 en un 44.5%, con lo cual esta configuración por sí sola no mejora la efectividad del clasificador; sin embargo, este filtro es importante para eliminar las palabras sin sentido dentro del texto.

Con la configuración adicional de IDFTransform y “TFTransform” en “True” , se obtiene un 91.1% en la precision de sus clases, mientras el recall también mejora a un 92.7%; así mismo la efectividad tanto a nivel de macro-f1 (49.3%) como del micro-f1 (91.2%), lo cual nos deja como conclusión mejora los resultados de la configuración anterior.

Incorporando el parámetro “normalizeDocLength” a “Normalize all data” en este método, se obtiene un 91.3% en la precision de sus clases, mientras el recall también mejora a un 92.7%; así mismo la efectividad tanto a nivel de macro-f1 (49.7%) como del micro-f1 (91.2%), mejoran considerablemente, transformandose en la mejor solución, ya que demuestra de acuerdo a nuestra métrica de evaluación macro-f1 su mejor valor.

Tabla N° 12 Filtros efectividad vs baseline

Preproceso filtros	Macro-F1	vs baseline	
por defecto	34,3%	1,0%	mejor
por defecto+tokenización	44,5%	11,2%	mejor
por defecto+tokenización+stopwords	44,5%	11,2%	mejor
por defecto + tokenización + stopwords +TF IDF	49,3%	16,0%	mejor
por defecto + tokenización + stopwords +TF IDF + normalizeDocLength	49,7%	16,4%	mejor

Con podemos evidenciar en la Tabla N° 12, se ha conseguido mejorar en 16.4 puntos con respecto a la línea base (33.3%) definida en el presente estudio; con lo cual de acuerdo al último experimento de filtros continuaremos con la siguiente fase en la experimentación de los clasificadores.

Finalmente, una vez aplicados los filtros tenemos que el archivo del conjunto de datos, ha sido modificado en su estructura para su posterior experimentación con los clasificadores. A continuación podemos apreciar la nueva estructura:

@relation 'NotiTrain-weka.filters.unsupervised.attribute.StringToWordVector-R1-W1000-prune-rate-1.0-T-I-N1-stemmerweka.core.stemmers.NullStemmer-stopwords-handlerweka.core.stopwords.WordsFromFile -stopwords C:\\Weka-3-8-M1-tokenizerweka.core.tokenizers.WordTokenizer -delimiters \" \\r \\t.,;:\\\'\\\"()?!/\\\\<>¿\"'

@attribute Tipo {0,1,2}

@attribute $ numeric

@attribute '\%' numeric

@attribute - numeric

……..

……..

@attribute vocalías numeric

@attribute voluntad numeric

@attribute voz numeric

@attribute Ángel numeric

@attribute ético numeric

@attribute única numeric

@attribute únicamente numeric

@attribute único numeric

@data

{22 2.812685,91 3.016478,94 3.176275,145 3.56893,162 1.208561,279 0.244378,390 1.584372,416 0.01837,418 2.480176,430 0.223463,474 3.585383,480 0.136682,489 0.185739,493 2.206718,619 0.070955,623 0.428492,641 0.39367,687 3.660974,688 2.760834,725 1.862185,737 0.465553,778 2.234751,828 3.289115,829 3.060516,834 0.203897,877 2.576103,922 1.308989,956 3.501386,965 2.604016,968 0.769878,970 3.046486,987 3.625176,993 0.244687,1442 7.452687}

{80 0.65952,84 2.04103,95 1.714138,98 1.618627,112 0.739079,162 0.742969,191 0.483134,268 1.904647,279 0.150233,298 1.731655,306 2.23582,309 1.732931,350 1.938706,376 0.802861,379 0.458208,382 2.174407,416 0.011293,417 1.617621,430 0.137375,434 1.65091,465 1.203019,480 0.084026,483 1.351944,489 0.114184,502 0.683069,503 1.603753,504 2.155565,521 1.554603,537 1.760541,571 1.973254,619 0.04362,623 0.263418,635 2.293771,641 0.242011,673 2.045907,692 1.944625,737 0.286201,742 1.801482,745 2.114278,776 1.279626,777 1.341084,781 0.345,783 2.200738,800 1.671825,834 0.125347,860 1.964899,884 0.25327,917 1.372013,926 1.969058,932 2.128619,940 1.393588,947 1.871137,953 1.89381,968 0.473287,969 0.478514,993 0.150422,1113

3.167034,1115 3.193008,1162 3.118952,1341 2.998218,1772 4.248582,1781 4.053792,1885 3.915586,1895 4.248582}

……..

……..

……..

6.2 Resultados exploración del clasificador LibSVM

Luego del análisis y optimización de los parámetros críticos de este clasificador, a través de la herramienta CVParameters, se obtuvieron los siguientes resultados; en los cuales se evidencia los valores de la matriz de confusión y de las métricas precision, recall y f1, así como el micro-f1 y macro-f1:

Tabla N° 13 Resultados filtros + (LibSVM por defecto)

Preproceso Desarrollo	neutral	positiva	negativa		instancias	Precision	Recall	f1
filtros + (por defecto)	9258	87	0	neutral	9345	93,2%	99,1%	96,1%
	577	377	0	positiva	954	81,1%	53,1%	53,1%
	95	1	0	negativa	96	0,0%	0,0%	0,0%

Micro-f1	**91,2%**
Macro-f1	49,7%

Tabla N° 14 Resultados filtros + (LibSVM por defecto + G)

Preproceso Desarrollo	neutral	positiva	negativa		instancias	Precision	Recall	f1
filtros + (por defecto + G)	9281	64	0	neutral	9345	92,1%	99,3%	95,6%
	698	256	0	positiva	954	79,8%	26,8%	40,2%
	95	1	0	negativa	96	0,0%	0,0%	0,0%

Micro-f1	**89,6%**
Macro-f1	45,3%

Tabla N° 15 Resultados filtros + (LibSVM por defecto + C)

Preproceso Desarrollo	neutral	positiva	negativa		instancias	Precision	Recall	f1
filtros + (por defecto + C)	9153	192	0	neutral	9345	94,6%	97,9%	96,3%
	432	522	0	positiva	954	72,6%	54,7%	62,4%
	89	5	2	negativa	96	100,0%	2,1%	4,1%

Micro-F1	**92,3%**
Macro-F1	54,3%

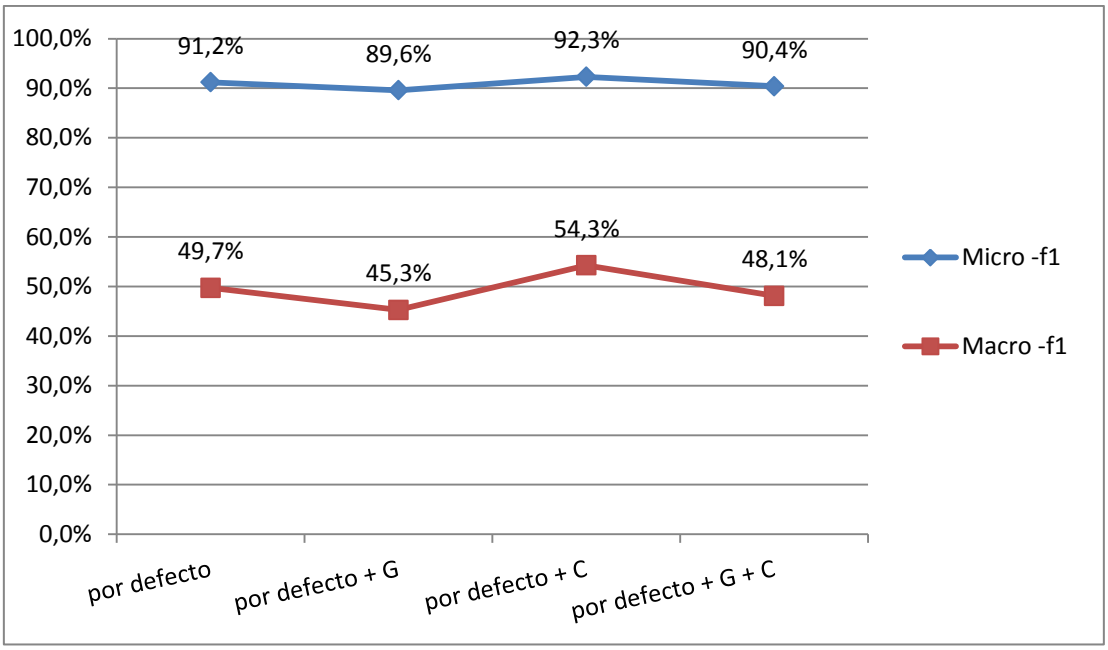
Tabla N° 16 Resultados filtros + (LibSVM por defecto + G + C)

Preproceso Desarrollo	neutral	positiva	negativa		instancias	Precision	Recall	f1
filtros + (por defecto + G + C)	9273	72	0	neutral	9345	92,6%	99,2%	95,8%
	644	310	0	positiva	954	80,9%	32,5%	46,4%
	94	1	1	negativa	96	100,0%	1,0%	2,1%

Micro-F1	**90,4%**
Macro-F1	48,1%

Para el análisis de resultados se ha generado la siguiente gráfica que resume la efectividad del clasificador.

Gráfica 5: LibSVM efectividad del clasificador



Con las mejores configuraciones determinadas en la sección exploración de filtros, y con las configuraciones de los parámetros por defecto de este clasificador, se puede ver una precision, recall y f1 del 0.00% en la clase 2 (negativa), con lo cual la efectividad del clasificador con esta configuración es del 91.2% de acuerdo al micro-f1, mientras que el 49.7% de rendimiento de acuerdo al macro-f1, en virtud de que la clase 2 (noticias negativas) está desbalanceada, pero que no interviene de manera notoria en esta configuración para mejorar el macro-f1.

Desarrollada la optimización del parámetro de gamma del núcleo (G) y obtenido su mejor valor (0.01) de configuración, y de acuerdo a las mejores configuraciones determinadas en la sección de exploración de filtros, se obtuvo una disminución en los valores de la efectividad del clasificador llegando a un 89.6% de acuerdo al micro-f1, ya que la precision, recall y f1 de la clase 1 (noticias positivas) se ha disminuido sus valores; así mismo se evidencia un disminución al 45.3% de rendimiento de acuerdo al macro-f1, y se mantienen

constante que la clase 2 (noticias negativas) no mejora sus valores del 0.0%.

Una vez realizada la optimización del parámetro coste (C) y obtenido su mejor valor (8) de configuración, se puede evidenciar una mejora sustancial en la clasificación de las clases de acuerdo a su tipo; la clase 1 (noticias positivas) mejora su recall y f1 considerablemente con respecto a la configuración anterior; adicional la clase 2 (noticias negativas) ya es tomada en cuenta con esta configuración y por ende mejora sus valores de precision, recall y f1; teniendo así una efectividad del clasificador del 92.3% de acuerdo al micro-f1, mejorando su valor con relación a la anterior; adicional existe un incremento al 54.3% de rendimiento de acuerdo al macro-f1; en tal virtud de acuerdo a esta configuración se puede evidenciar que ha mejorado su efectividad.

Para finalizar la experimentación con este clasificador, se realiza la parametrización con la mejor configuración de los filtros y la conjugación de los parámetros del clasificador optimizados (C + G), evidenciando que no existe mejora con respecto a las configuraciones individuales anteriores; en cuanto a la efectividad a nivel del micro-f1, obtiene un 90.4%, y el macro-f1 obtiene un resultado del 48.1%, con lo que se refleja que existe una disminución con relación al valor encontrado en el experimento anterior; con lo cual esta no mejora las configuraciones por defecto del clasificador, ni la individual del parámetro (C).

Tabla N° 17 LibSVM efectividad vs baseline

Preproceso Desarrollo	Macro -f1	vs baseline	
por defecto	49,7%	16,4%	mejor
por defecto + G	45,3%	12,0%	mejor
por defecto + C	54,3%	21,0%	mejor
por defecto + G + C	48,1%	14,8%	mejor

De acuerdo a la tabla N° 17, de los resultados y de acuerdo al análisis, para la siguiente etapa de experimentación con el conjunto de datos test, se lo realizará en virtud de la configuración que ha obtenido los resultados de mejor efectividad de acuerdo al macro-f1; es decir, el clasificador LibSVM con su parámetro optimizado (C), que alcanza un 54.3% de efectividad, y teniendo en cuenta que ha mejorado 21 puntos en relación a la línea base.

6.3 Resultados exploración del clasificador NaiveBayes

Luego del análisis de parámetros y configuraciones se han conseguido los siguientes resultados, en los cuales de muestra los valores de la matriz de confusión y de las métricas de evaluación precision, recall y f1 por cada clase, así como la efectividad del clasificador de acuerdo al micro-f1 y macro-f1:

Tabla N° 18 Resultados filtros + NaiveBayes por defecto

Preproceso Desarrollo	neutral	positiva	negativa		instancias	Precision	Recall	f1
Filtros + por defecto	6118	565	2662	neutral	9345	97,3%	65,5%	78,3%
	129	668	157	positiva	954	53,7%	70,0%	60,8%
	38	10	48	negativa	96	1,7%	50,0%	3,2%

Micro-f1	**76,0%**
Macro-f1	47,4%

Tabla N° 19 Resultados filtros + NaiveBayes por defecto + D

Preproceso Desarrollo	neutral	positiva	negativa		instancias	Precision	Recall	f1
Filtros + por defecto + D	8567	521	257	neutral	9345	96,0%	91,7%	93,8%
	300	646	8	positiva	954	54,3%	67,7%	60,3%
	54	22	20	negativa	96	7,0%	20,8%	10,5%

Micro-f1	**90,0%**
Macro-f1	54,9%

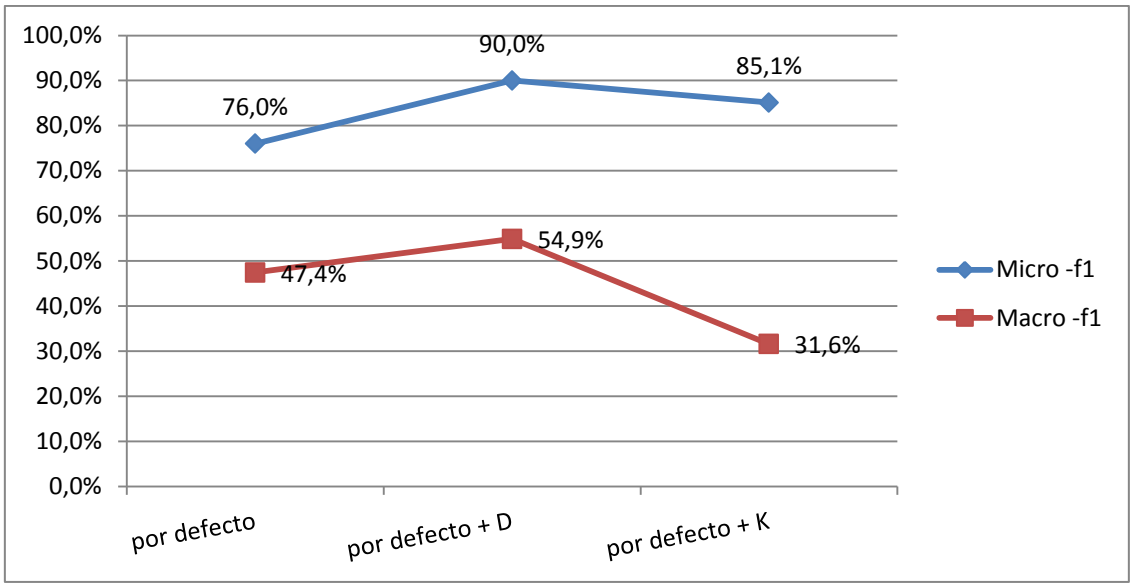
Tabla N° 20 Resultados filtros + NaiveBayes por defecto + K

Preproceso Desarrollo	neutral	positiva	negativa		instancias	Precision	Recall	f1
Filtros + por defecto + K	9345	0	0	neutral	9345	89,9%	100,0%	94,7%
	954	0	0	positiva	954	0,0%	0,0%	0,0%
	96	0	0	negativa	96	0,0%	0,0%	0,0%

Micro-f1	**85,1%**
Macro-f1	31,6%

Para el análisis de resultados se han generado las siguientes gráficas que resumen la clasificación y la efectividad del clasificador.

Gráfica 6: NaiveBayes efectividad del clasificador



Con las mejores configuraciones determinadas en la exploración de filtros, y con las configuraciones por defecto de este clasificador, la efectividad del clasificador con esta configuración es del 76.0% de acuerdo al micro-f1, mientras que el 47.4% de rendimiento de acuerdo al macro-f1, con lo que se demuestra que disminuye 2.3 puntos en esta métrica, y por ende su efectividad con respecto al experimento del mejor filtro encontrado. Cabe mencionar que la clase 2 (noticias negativas) si obtiene mejores resultados en su precision, recall y f1, que el anterior clasificador (LibSVM) en su configuración por defecto.

Utilizando el parámetro (D) y las mejores configuraciones determinadas en la exploración de filtros, la efectividad del clasificador con esta configuración mejora considerablemente con relación a la configuración por defecto, obteniendo un 90.30% de acuerdo al micro-f1; así mismo se evidencia un incremento al 54.9% de rendimiento de acuerdo al macro-f1; en tal virtud de acuerdo a esta configuración la clase 2 (noticias negativas) minoritaria mejora considerablemente sus valores de clasificación precision, recall y f1, con lo cual se demuestra que la efectividad del clasificador mejora en su macro-f1, incluso

mejora 0.6 puntos con relación al anterior clasificador (LibSVM).

Al utilizar el kernel estimator (K) y de acuerdo a las mejores configuraciones determinadas en la exploración de filtros, se refleja una mejora mínima en la clasificación de las clases; adicional la efectividad del clasificador con esta configuración es del 85.1% de acuerdo al micro-f1, disminuyendo su valor con relación a la anterior configuración, y se evidencia un disminución al 31.6% de rendimiento de acuerdo al macro-f1; adicional de acuerdo a los datos del resultado, se puede concluir que el clasificador no está realizando su clasificación correcta, ya que las clases 1 (noticias positivas) y 2 (noticias negativas) reflejan valores de 0.0% en sus métricas de precision, recall y f1.

Tabla N° 21 NaiveBayes efectividad vs baseline

Preproceso Desarrollo	Macro -f1	vs baseline	
por defecto	47,4%	14,1%	mejor
por defecto + D	54,9%	21,6%	mejor
por defecto + K	31,6%	-1,7%	peor

Con estos resultados, de acuerdo a la table N° 21, para la siguiente etapa de experimentación con el conjunto de datos test, se lo realizará con la configuración de mejor efectividad que alcanza el 54.9% de acuerdo al macro-f1, es decir el clasificador NaiveBayes con su parámetro (D), ya que existe una mejora con relación a la línea base de 21.6 puntos porcentuales.

6.4 Resultados exploración del clasificador J48

Una vez configurados los parámetros por defecto y parámetros críticos optimizados, se obtuvieron los siguientes resultados, en los cuales se evidencia los valores de la matriz de confusión y de las métricas precision, recall y f1 por

cada clase, así como la efectividad del clasificador de acuerdo al micro-f1 y macro-f1:

Tabla N° 22 Resultados filtros + (J48 por defecto)

Preproceso Desarrollo	neutral	positiva	negativa		instancias	Precision	Recall	f1
filtros + (por defecto)	9104	234	7	neutral	9345	94,7%	97,4%	96,0%
	443	507	4	positiva	954	67,2%	53,1%	59,3%
	67	14	15	negativa	96	57,7%	15,6%	24,6%

Micro-f1	**92,0%**
Macro-f1	60,0%

Tabla N° 23 Resultados filtros + (J48 por defecto + C)

Preproceso Desarrollo	neutral	positiva	Negativa		instancias	Precision	Recall	f1
filtros + (por defecto + C)	9134	204	7	neutral	9345	94,7%	97,7%	96,2%
	449	501	4	positiva	954	69,6%	52,5%	59,9%
	67	15	14	negativa	96	56,0%	14,6%	23,1%

Micro-f1	**92,2%**
Macro- f1	59,7%

Tabla N° 24 Resultados filtros + (J48 por defecto + M)

Preproceso Desarrollo	neutral	positiva	negativa		instancias	Precision	Recall	f1
filtros + (por defecto + M)	9148	193	4	neutral	9345	94,4%	97,9%	96,1%
	468	483	3	positiva	954	69,9%	50,6%	58,7%
	74	15	7	negativa	96	50,0%	7,3%	12,7%

Micro- f1	**91,9%**
Macro- f1	55,8%

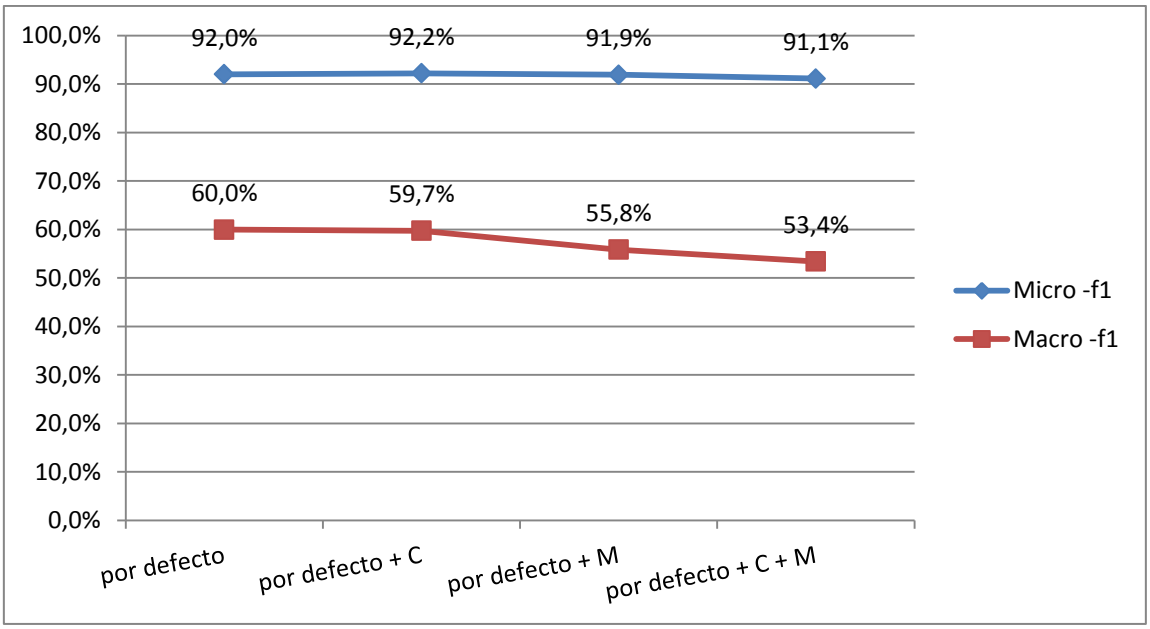
Tabla N° 25 Resultados filtros + (J48 por defecto + M + C)

Preproceso Desarrollo	neutral	positiva	negativa		instancias	Precision	Recall	f1
filtros + (por defecto + M + C)	9235	106	4	neutral	9345	93,3%	98,8%	96,0%
	582	369	3	positiva	954	76,9%	38,7%	51,5%
	84	5	7	negativa	96	50,0%	7,3%	12,7%

Micro- f1	**91,1%**
Macro- f1	53,4%

Para el análisis de resultados se ha generado la siguiente gráfica que resume la clasificación y la efectividad del clasificador.

Gráfica 7: J48 efectividad del clasificador



Con las mejores configuraciones determinadas en la exploración de filtros, y con las configuraciones por defecto de este clasificador, la efectividad del clasificador con esta configuración es del 92.0% de acuerdo al micro-f1, mientras que el 60.0% de rendimiento de acuerdo al macro-f1, en virtud de que la clase 3 (noticias negativas) está desbalanceada, pero que está siendo muchos más considerada y clasificada correctamente en este experimento, con lo cual, de acuerdo a esta configuración ha mejorado sustancialmente la efectividad de acuerdo al macro-f1, ya que se incrementa 10.3 puntos de porcentaje con relación a la mejor configuración en la exploración de filtros, se incrementa 5.7 puntos porcentuales con relación al mejor valor del clasificador LibSVM, y se incrementa 5.1 puntos porcentuales con relación al mejor valor del clasificador NaiveBayes.

Una vez realizado la optimización del parámetro factor de confianza (C) y obtenido su mejor valor (0.1) de configuración, se evidencia que la efectividad del clasificador con esta configuración es del 92.2% de acuerdo al micro-f1, mejorando su valor con relación al anterior experimento; sin embargo, se

evidencia una mínima disminución de 0.3 puntos porcentuales de rendimiento, es decir tenemos un 59.7% de acuerdo al macro-f1; adicional los valores de precision, recall y f1 de las clases 1 (noticias positivas) y clase 2 (noticias negativas) han disminuido, con lo cual se se ha afectado a la efectividad del clasificador con esta configuración.

Desarrollada la optimización del parámetro de instancias por hoja del árbol (M) y obtenido su mejor valor (5) de configuración, se puede deducir que la efectividad, de acuerdo al micro-f1 ha disminuido su valor con relación a la anterior configuración; así mismo, se evidencia una disminución al 55.8% de rendimiento de acuerdo al macro-f1, debido a que las clases desbalanceadas y aún más la clase minoritaria disminuyen sus valores en las métricas de evaluación precision, recall y f1.

Finalmente se realiza la parametrización con la mejor configuración de los filtros y los parámetros del clasificador optimizados (C + M), evidenciando que no existe mejora sustancial con relación a sus configuraciones anteriores, ya que la efectividad disminuye tanto a nivel del micro-f1, obteniendo un 91.1%, así como el macro-f1 que obtiene un resultado del 53.4%.

Tabla N° 26 J48 efectividad vs baseline

Preproceso Desarrollo	Macro -f1	vs baseline	
por defecto	60,0%	26,7%	mejor
por defecto + C	59,7%	26,4%	mejor
por defecto + M	55,8%	22,5%	mejor
por defecto + C + M	53,4%	20,1%	mejor

Una vez analizados estos resultados, y de acuerdo a la table N° 26, para la siguiente etapa de experimentación con el conjunto de datos test, se lo realizaría en virtud de la configuración que ha obtenido los resultados de mejor efectividad

de acuerdo al micro-f1 que obtiene un 60.0% de efectividad, es decir el clasificador J48 con sus parámetros por defecto. Además, estos resultados presentan una mejora de 26.7 puntos porcentuales con respecto a la línea base.

6.5 Resultados conjunto de datos test

Como es posible que haya algo de sobreajuste con el conjunto de datos de desarrollo, nuestras medidas finales deben tomarse con un conjunto de datos independiente que no se usó en el entrenamiento; para lo cual el conjunto de datos test, nos permitirá probar y evaluar el modelo con las mejores configuraciones obtenidas en los experimentos anteriores con los datos de desarrollo, es decir, una vez que se han encontrado las mejores configuraciones en el proceso de exploración de filtros, y en la exploración de los clasificadores LibSVM, NaiveBayes y J48 de acuerdo a sus parámetros, procedemos a realizar la experimentación sobre el conjunto de datos test, obteniendo los siguientes resultados, en los cuales se puede observar los valores de la matriz de confusión y de las métricas de evaluación precision, recall y f1 por cada clase, así como la efectividad del clasificador de acuerdo al micro-f1 y macro-f1:

Tabla N° 27 Resultados test LibSVM

Preproceso TEST	neutral	positiva	negativa		instancias	Precision	Recall	f1
LibSVM	9158	185	2	neutral	9345	94,7%	98,0%	96,3%
	429	525	0	positiva	954	72,8%	55,0%	62,7%
	84	11	1	negativa	96	33,3%	1,0%	2,0%

Micro-f1	**92,4%**
Macro-f1	53,7%

Tabla N° 28 Resultados test NaiveBayes

Preproceso TEST	neutral	positiva	negativa		instancias	Precision	Recall	f1
NaiveBayes	8793	455	97	neutral	9345	95,9%	94,1%	95,0%
	312	637	5	positiva	954	56,9%	66,8%	61,5%
	66	27	3	negativa	96	2,9%	3,1%	3,0%

Micro-f1	**91,1%**
Macro-f1	53,2%

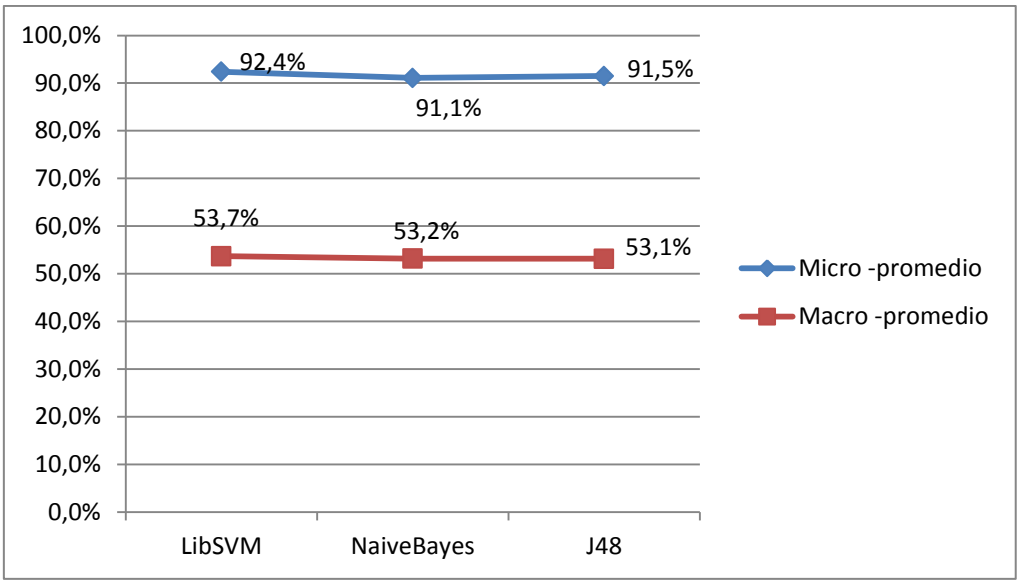
Tabla N° 29 Resultados test J48

Preproceso TEST	neutral	positiva	negativa		instancias	Precision	Recall	f1
J48	9038	296	11	neutral	9345	94,7%	96,7%	95,7%
	428	523	3	positiva	954	62,6%	54,8%	58,4%
	76	17	3	negativa	96	17,6%	3,1%	5,3%

Micro-f1	**91,5%**
Macro- f1	53,1%

Para el análisis de resultados se ha generado la siguiente gráfica que resume la efectividad del clasificador.

Gráfica 8: Efectividad clasificadores conjunto de datos test



Luego de la fase de exploración de clasificadores, se ha procedido a realizar la experimentación con el conjunto de datos test, obteniendo los siguientes resultados con el clasificador LibSVM, en los cuales la efectividad del clasificador es del 92.4% de acuerdo al micro-f1, mientras que el 53.7% de rendimiento de acuerdo al macro-f1; con lo cual se puede evidenciar que si bien la efectividad es aceptable y equilibrada, su valor f1 es bajo en la clase 3 (noticias negativas), que es la clase desbalanceada y de importante consideración para la evaluación integral en el presente estudio.

Continuado la experimentación del conjunto de datos test con el clasificador NaiveBayes, obtenemos los siguientes resultados en la efectividad del clasificador, con esta configuración es del 91.1% de acuerdo al micro-f1;, sin embargo, se evidencia una disminución mínima al 53.2% de rendimiento de acuerdo al macro-f1; mejora en un punto porcentual la clasificación de la clase minoritaria en su f1.

Finalmente, experimentamos el conjunto de datos test con el clasificador J48 y su mejor configuración de parámetros, evaluada en el proceso de exploración del clasificador con el conjunto de datos de desarrollo, obtenemos que la

efectividad del clasificador es del 91.5% de acuerdo al micro-f1, mejorando su valor en 0.4 puntos porcentuales con relación a la anterior configuración, y un 53.1% de rendimiento de acuerdo al macro-f1; sin embargo, si bien los valores del resultado obtenidos son menores con relación al clasificador LibSVM, se puede apreciar que para la clase 3 (noticias negativas), la más desbalanceada del conjunto de datos, mejora la efectividad individual de acuerdo a la métrica f1 en un 2.3 puntos porcentuales en relación con el clasificador LibSVM.

Tabla N° 30 Test efectividad vs baseline

Preproceso TEST	Macro -f1	vs baseline	
LibSVM	53,7%	20,4%	mejora
NaiveBayes	53,2%	19,9%	mejora
J48	53,1%	19,8%	mejora

Como se evidencia en la tabla N° 30, podemos concluir que el clasificador LibSVM es el mejor clasificador ya que mejora 20.4 puntos porcentuales con relación a la línea base; sin embargo hay que considerar que el clasificador J48 mejora la efectividad individual de la clase minoritaria, por consiguiente la definición final del mejor clasificador, se determinará en la siguiente sección, una vez que valide el sobreajuste.

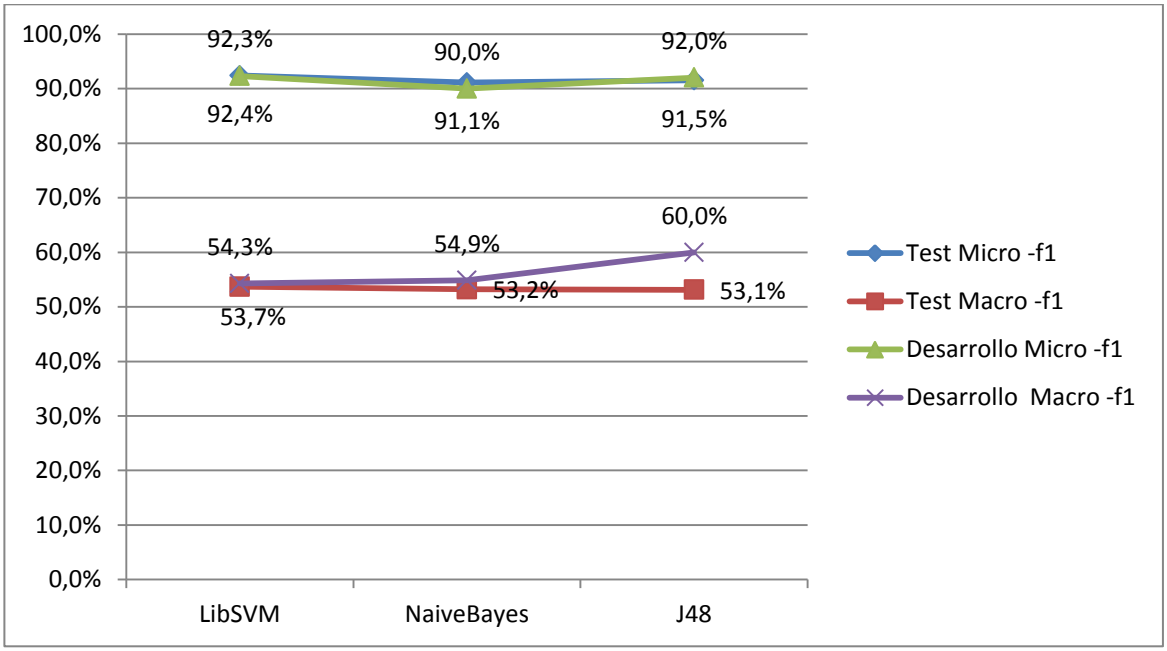
6.6 Resultados conjunto de datos desarrollo vs test

Para realizar el análisis en este tema, es importante contextualizar el concepto de sobreajuste (overfitting), que es el efecto de sobreentrenar un algoritmo de aprendizaje con unos ciertos datos para los que se conoce el resultado deseado. El algoritmo de aprendizaje debe alcanzar un estado en el que será capaz de predecir el resultado en otros casos a partir de lo aprendido con los datos de entrenamiento, generalizando para poder resolver situaciones distintas a las

acaecidas durante el entrenamiento. Sin embargo, cuando un sistema se entrena demasiado (se sobreentrena) o se entrena con datos extraños, el algoritmo de aprendizaje puede quedar ajustado a unas características muy específicas de los datos de entrenamiento que luego no sabe generalizar con el conjunto de datos test. (Castillo González, 2015)

Con lo anteriormente expuesto, el análisis de los resultados con los datos desarrollo versus los datos del conjunto test, tenemos en la siguiente gráfica:

Gráfica 9: Efectividad clasificadores conjunto de datos test vs desarrollo



En la gráfica, se puede apreciar que el micro-f1 en los clasificadores LibSVM, NaiveBayes y J48 con los datos desarrollo y test son similares con una variación mínima de 0.1 a 1.1 puntos porcentuales, por cuanto no existe un sobreajuste considerable; así mismo, en el macro-f1 del clasificador LibSVM existe una variación mínima de 0.6 puntos porcentuales, mientras que en el clasificador NaiveBayes existe una disminución en el macro-f1 de 1.7 puntos porcentuales, en cambio en el clasificador J48 existe una disminución en el macro-f1 de 6.8 puntos porcentuales, lo cual evidencia un sobreajuste considerable del clasificador J48 con el conjunto de datos test.

7. CONCLUSIONES

Una vez aplicadas las técnicas de minería de texto, con un conjunto de datos desbalanceado, utilizando una herramienta open source (weka) y aplicando una metodología ágil y sencilla, hemos logrado mejorar los resultados de la clasificación de la polaridad del conjunto de datos, a través de los diferentes experimentos y configuraciones implementadas en el presente estudio, proporcionando un modelo robusto y estable para este tipo de problemas de clasificación.

Por lo tanto se pueden establecer las siguientes conclusiones específicas:

- La minería de textos, y la clasificación de la polaridad de documentos en particular, son un campo de investigación y aplicación prometedor; pues cada día los organismos están orientando sus esfuerzos para aprovechar el gran volumen de conocimiento no estructurado que disponen.
- La clasificación automática de documentos utilizando el modelo aprendido es de suficiente calidad comparada con la clasificación realizada manualmente. En base a los experimentos realizados, el algoritmo LibSVM (máquinas de vectores soporte) tiene un rendimiento bastante estable en cuanto a los resultados de la clasificación de acuerdo al macro-f1.
- Los modelos aprendidos por el algoritmo J48 son muy buenos también para el problema planteado, sin embargo, en este clasificador se produce un sobreajuste en el macro-f1 al clasificar los documentos con un conjunto de datos (test) independiente.
- De acuerdo a nuestra métrica seleccionada en este presente estudio, se ha determinado que de acuerdo a su efectividad en el macro-f1 es el mejor clasificador es LibSVM, ya que adicionalmente no tendría sobreajuste considerables al cambiar de conjunto de datos.
- Se ha mejorado la línea base definida en 20.4 puntos porcentuales, con el experimento del conjunto de datos test y del mejor clasificador (LibSVM)

identificado en este estudio, es decir hemos alcanzado una efectividad del 53.7% de acuerdo al macro-f1.

- Las experiencias presentadas muestran el impacto en la efectividad de los clasificadores ante la variación de parámetros de entrada, tipos de atributos extraídos, conjunto de datos de entrenamiento, conjunto de datos de desarrollo, conjunto de datos de test, y proporción de documentos de cada clase.
- Dado que no existen trabajos con los cuales compararnos directamente, ha sido necesario establecer una línea base (baseline), con la cual, se tiene una referencia acerca de la validez de los resultados en los experimentos de clasificación automática.

8. RECOMENDACIONES

- Para un trabajo futuro se recomienda experimentar con conjuntos de datos balanceados en las diferentes clases, y validar la efectividad del mejor clasificador identificado en el presente estudio.

- Se recomienda realizar una experimentación en la cual se valide si es representativa la mejora en conjuntos de datos mayores o menores instancias a los planteados en el presente estudio.

- Se plantea la posibilidad de que en otros trabajos se analice otros campos de la estructura inicial del conjunto de datos, como por ejemplo el tipo de medio de comunicación (prensa escrita, radio, televisión y medio web).

- Se recomienda para posteriores estudios de análisis, identificar cuáles son las palabras o características más importantes al clasificar los documentos o las noticias en positivas, negativas o neutrales.

- En el presente estudio se analizó la clasificación de polaridad como una de las técnicas de la minería de texto, por lo que sería importante contrastar los resultados y profundizar en las demás técnicas, como por ejemplo el procesamiento del lenguaje natural.

9. BIBLIOGRAFÍA

Camelo, S., Bernal, M., & Barreras, F. (Agosto 2015). Minería de Texto. Minería de Texto.

Castillo González, N. (Septiembre 2015). Técnicas de Machine Learning para el Post-Proceso. Técnicas de Machine Learning para el Post-Proceso.

Ian, W., Eibe, F., & Mark, H. (2011). DATA MINING Practical Machine Learning Tools and Technique. Morgan Kaufmann Publishers is an imprint of Elsevier.

Optimizing parameters. (s.f.). Obtenido de https://weka.wikispaces.com/Optimizing+parameters

Rainbow. (Septiembre de 1998). Obtenido de Rainbow: http://www.cs.cmu.edu/~mccallum/bow/rainbow

Castillo, G., Nuria, V. (Septiembre 2015). Técnicas de Machine Learning para el Post-Proceso de la predicción de la Irradiancia.

Dubiau, Luciana. (Octubre 2013). Procesamiento de Lenguaje Natural en

Sistemas de Análisis de Sentimientos.

Ortega, C., Walter, R. (Marzo 2015). Aplicación de técnicas de procesamiento de Lenguaje Natural y Minería de Texto para la clasificación de preguntas dentro de un cuestionario digital.

Ian H. Witten. (2014). Data Mining with Weka: weka.waikato.ac.nz

Cotelo, M., Juan, M. (Junio 2015). Análisis de Contenidos Generados por Usuarios mediante la Integración de Información Estructurada y No Estructurada.

Díaz, R.,Ismael (Enero 2013). Detección de afectividad en texto en español

basada en el contexto ligüístico para síntesis de voz.

Clasificación de Texto: medidas de rendimiento y desempeño. Obtenido de:

http://pdln.blogspot.com/2013/06/clasificacion-de-texto-medidas-de.htmls

Aplicación de Algoritmos de Clasificación de Minería de Textos para el Reconocimiento de Habilidades de E-tutores Colaborativos. Obtenido de Revista Iberoamericana de Inteligencia Artificial:

http://www.redalyc.org/pdf/925/92530455007.pdf

Text Categorization with Class-Based and Corpus-Based Keyword Selection. Obtenido de:

http://www.cmpe.boun.edu.tr/~gungort/papers/Text%20Categorization%20with%20Class-Based%20and%20Corpus-Based%20Keyword%20Selection.pdf

Clasificación de Documentos usando Naive Bayes Multinomial y Representaciones Distribucionales. Obtenido de:

https://ccc.inaoep.mx/~esucar/Clases-mgp/Proyectos/2011/Reporte_Proyecto_Clasificacion_de_Documentos.pdf

Desarrollo de clasificadores ensamblados robustos ante el problema de clases desbalanceadas en la clasificación multinstancia. Obtenido de:

https://www.researchgate.net/publication/281243241_Desarrollo_de_clasificlasifi_ensamblados_robustos_ante_el_problema_de_clases_desbalanceadas_en_la_clasificacion_multinstancia