

UNIVERSIDAD DEL BÍO-BÍO

Facultad de Ciencias Empresariales
Departamento Sistema de Informaciones



UNIVERSIDAD DEL BÍO-BÍO

MEMORIA PARA OPTAR A TÍTULO DE INGENIERO DE EJECUCIÓN EN
COMPUTACIÓN E INFORMÁTICA

**Detección de violencia verbal hacia las mujeres en redes sociales
mediante técnicas de aprendizaje automático.**

Alumno: Ricardo A. Riquelme Silva.

Profesor Guía: Dr. Christian L. Vidal Castro.

CONCEPCION, 2019

RESUMEN

La llegada de la web y la creación de redes sociales como Twitter, ha generado un espacio donde se reciben muchas opiniones, entre ellas, proliferan actitudes de usuarios que expresan sin pudor conductas agresivas en contra de la mujer. Como una posible solución a este problema y evitar que ello se transforme en violencia, en este trabajo se utiliza Machine Learning para la detección de agresividad, para lo cual se construyó un corpus y se utilizaron técnicas para la detección, que arrojaron resultados de buen nivel de rendimiento y certeza en la predicción. Todo lo cual permite iniciar un camino hacia la utilización de técnicas del análisis de sentimiento, para la detección de agresividad hacia la mujer en comentarios de redes sociales.

Palabras clave – Análisis de sentimiento, Aprendizaje automático, Rendimiento de clasificadores supervisados

ABSTRACT

The arrival of the web and the creation of social networks such as Twitter, has become a space where there are many opinions, among them the attitudes of users who shamelessly express aggressive behaviors against women have multiplied. As a solution, this problem and prevent it from turning into violence, in this work is used Automatic learning for the detection of aggressiveness, for which a corpus was built and detection techniques were used, which yielded results of good performance level and certainty in the prediction. All of which allows to start a path towards the use of sentiment analysis techniques, for the detection of aggressiveness towards women in comments from social networks.

Keywords - Sentiment analysis, Machine learning, Performance of supervised classifiers

INDICE DE CONTENIDOS

INDICE DE CONTENIDOS	1
INDICE DE ILUSTRACIONES	3
INDICE DE TABLAS.....	4
INDICE DE FORMULAS.....	5
1. INTRODUCCIÓN.....	6
1.1. Violencia contra la mujer en Chile y el mundo.	6
1.2. Machine Learning.	8
2. PERFIL DEL PROYECTO	10
2.1. Objetivo General.	10
2.2. Objetivos Específicos.....	10
2.3. Alcances de la Investigación.	10
2.4. Metodología.	11
3. MARCO TEORICO	13
3.1. Análisis del Problema de Violencia contra la mujer en Chile.	13
3.2. Machine Learning.	15
3.3. Clasificación de Métodos de Machine Learning.	17
3.3.1. Mediante aprendizaje supervisado.....	17
3.3.2. Mediante aprendizaje no supervisado.	18
3.4. Preparación de los datos: limpieza del corpus.	18
3.5. Tokenización.....	20
3.6. Reducción de las características.	21
3.7. Features etiquetado POS.	22
3.8. Stop words.	22
3.9. Software WEKA	23
3.10. Técnicas y Algoritmos.....	25
3.10.1. Clasificador de Naive Bayes.	25
3.10.2. Clasificador Árboles de Decisión (Decision Tree - Algoritmo C4.5 (J48)). ..	26
3.10.3. Clasificador Random Forest.....	27
3.10.4. Clasificador Máquinas de Vectores de Soporte (SVM).	28

Ingeniería de Ejecución en Computación e Informática - UBB

3.11.	Métricas y métodos de evaluación de resultados.	29
3.12.	K-fold Cross Validation.	32
3.13.	Proyectos Relacionados.	33
4.	DESARROLLO DEL MODELO.	36
4.1.	Recolección de Datos.	36
4.2.	Limpieza de Datos.	38
4.3.	Elección de Algoritmos.	38
4.4.	Creación del Corpus (Acuerdo Kappa).	39
4.5.	Descripción del Corpus.	44
4.6.	Preprocesamiento.	49
5.	EXPERIMENTACION	50
5.1.	Introducción.	50
5.2.	Descripción de los experimentos.	50
6.	RESULTADOS Y ANALISIS DE LAS EXPERIMENTACIONES	53
6.1.	Experimento N° 1.	53
6.2.	Experimento N° 2	56
7.	APLICACIÓN DESARROLLADA PARA USO DEL MODELO.	59
8.	CONCLUSIONES.	63
8.1.	Aportes.	63
8.2.	Contraste de lo realizado con los objetivos planteados.	64
8.3.	Conclusiones.	65
8.4.	Trabajos Futuros.	66
9.	REFERENCIAS.	68
10.	ANEXOS	71
	Anexo 1: Matrices de confusión para cálculo de Coeficiente Kappa de Cohen.	71
	Anexo 2: Listado de Stops Words usados en la limpieza de los Corpus.	73
	Anexo 3: Resultados de la clasificación entregados por WEKA.	75

INDICE DE ILUSTRACIONES

Ilustración 1: Diagrama con Metodología de Trabajo.	12
Ilustración 2: Esquema de Árbol de Decisión (Decision Tree).	27
Ilustración 3: Esquema de Bosque Aleatorio (Random Forest).	28
Ilustración 4: Esquema de Máquina de Soporte de Vectores (SVM).	29
Ilustración 5: Muestra aumento de interés en búsquedas en Google sobre el tema.	36
Ilustración 6: Diagrama con pares de evaluadores para coeficiente Kappa de Cohen.	41
Ilustración 7: Nube de palabras correspondiente al corpus final.	43
Ilustración 8: Gráfica muestra el balance del corpus.	43
Ilustración 9: Método para eliminar comillas y espacios dobles.	46
Ilustración 10: Método para reemplazar caracteres especiales.	46
Ilustración 11: Método para eliminar signos de puntuación.	47
Ilustración 12: Método para eliminar signos de exclamación, interrogación y otros.	47
Ilustración 13: Método para eliminar las Stop Words.	47
Ilustración 14: Ejemplo de funcionamiento de Stanford POS Tagger.	48
Ilustración 15: Diagrama con proceso etapa de experimentación.	52
Ilustración 16: Gráfico con rendimiento de clasificadores en Experimento N° 1.	55
Ilustración 17: Gráfico con rendimiento de clasificadores en Experimento N° 2.	58
Ilustración 18: Función que carga el corpus al programa.	59
Ilustración 19: Método que define % de entrenamiento y pruebas.	60
Ilustración 20: Función que realiza la vectorización de los datos.	60
Ilustración 21: Función que Tokeniza los textos ingresados.	60
Ilustración 22: Método aplica modelo SVM del módulo sklearn de biblioteca Pandas.	61
Ilustración 23: Método donde se aplica el modelo predictivo y su respectivo reporte.	61
Ilustración 24: Función que analiza nuevos comentarios y los clasifica.	61
Ilustración 25: Función para crear interfaz gráfica.	62
Ilustración 26: Ejemplo de interfaz gráfica generada por el programa.	62
Ilustración 27: Anexo Resultados Clasificación WEKA - Corpus 1 Clasificador DT.	75
Ilustración 28: Anexo Resultados Clasificación WEKA - Corpus 1 Clasificador NB.	75

Ingeniería de Ejecución en Computación e Informática - UBB

Ilustración 29: Anexo Resultados Clasificación WEKA - Corpus 1 Clasificador SVM.....	76
Ilustración 30: Anexo Resultados Clasificación WEKA - Corpus 1 Clasificador RF.....	76
Ilustración 31: Anexo Resultados Clasificación WEKA - Corpus 2 Clasificador DT	77
Ilustración 32: Anexo Resultados Clasificación WEKA - Corpus 2 Clasificador NB	77
Ilustración 33: Anexo Resultados Clasificación WEKA - Corpus 2 Clasificador SVM.....	78
Ilustración 34: Anexo Resultados Clasificación WEKA - Corpus 2 Clasificador RF.....	78
Ilustración 35: Anexo Resultados Clasificación WEKA - Corpus 3 Clasificador DT	79
Ilustración 36: Anexo Resultados Clasificación WEKA - Corpus 3 Clasificador NB	79
Ilustración 37: Anexo Resultados Clasificación WEKA - Corpus 3 Clasificador SVM.....	80
Ilustración 38: Anexo Resultados Clasificación WEKA - Corpus 3 Clasificador RF.....	80
Ilustración 39: Anexo Resultados Clasificación WEKA - Corpus 4 Clasificador DT	81
Ilustración 40: Anexo Resultados Clasificación WEKA - Corpus 4 Clasificador NB	81
Ilustración 41: Anexo Resultados Clasificación WEKA - Corpus 4 Clasificador SVM.....	82
Ilustración 42: Anexo Resultados Clasificación WEKA - Corpus 4 Clasificador RF.....	82

INDICE DE TABLAS

Tabla 1: Ejemplo de set de comentarios y etiquetado de evaluadores.....	38
Tabla 2: Rangos de acuerdo en coeficiente Kappa de Cohen.....	40
Tabla 3: Ejemplo de corpus con sus evaluadores y proceso de clasificación final.....	41
Tabla 4: Ejemplo de extracto de corpus resultante.	42
Tabla 5: Ejemplo de corpus antes y después de etiquetado POS.....	49
Tabla 6: Ejemplo de las distintas versiones de corpus.	50
Tabla 7: Rendimiento de clasificadores en Experimento N° 1.....	53
Tabla 8: Resultados de métricas Precision, Recall y F-Measure en Experimento N° 1.	54
Tabla 9: Resumen de matrices de confusión en Experimento N° 1.....	55
Tabla 10: Rendimiento de clasificadores en Experimento N° 2.....	56
Tabla 11: Resultados de métricas Precision, Recall y F-Measure en Experimento N° 2.	57
Tabla 12: Resumen de matrices de confusión en Experimento N° 2.....	58

Ingeniería de Ejecución en Computación e Informática - UBB

Tabla 13: Matriz de confusión (Evaluador 1 y Evaluador 2) índice Kappa.....	71
Tabla 14: Coeficiente Kappa (Evaluador 1 y Evaluador 2).....	71
Tabla 15: Matriz de confusión (Evaluador 2 y Evaluador 3) índice Kappa.....	71
Tabla 16: Coeficiente Kappa (Evaluador 2 y Evaluador 3).....	71
Tabla 17: Matriz de confusión (Evaluador 1 y Evaluador 3) índice Kappa.....	72
Tabla 18: Coeficiente Kappa (Evaluador 1 y Evaluador 3).....	72
Tabla 19: Promedio coeficiente Kappa de Cohen.....	72
Tabla 20: Rangos de concordancia en coeficiente Kappa de Cohen.	72
Tabla 21: Set con Stop Words utilizados en limpieza de los corpus.	73
Tabla 22: Continuación set de Stop Words utilizados en limpieza de los corpus.	74

INDICE DE FORMULAS

Ecuación 1: Clasificador Naive Bayes	25
Ecuación 2: Métrica Exactitud (Accuracy).	30
Ecuación 3: Métrica Precisión (Precision).	31
Ecuación 4: Métrica Exhaustividad (Recall).	31
Ecuación 5: Métrica Valor-F (F-Measure).	32
Ecuación 6: Definiciones matemáticas de coeficiente Kappa de Cohen.....	40

1. INTRODUCCIÓN

El sustancial crecimiento en la utilización de redes sociales, como medio en que millones de personas expresan opiniones y comentarios de temas generales y particulares que puedan afectar a gobiernos, organizaciones, instituciones, empresas y personas, ciertamente resultan en una muy interesante y válida fuente de información, que permite mediante procesos informáticos automatizados, detectar ciertas actitudes en el comportamiento de las personas, en este caso particular la agresividad en contra de la mujer. Es por lo que el análisis de sentimiento en los comentarios publicados en Twitter a través del aprendizaje automático supervisado, nos permitirá analizar un gran volumen de textos a partir de un corpus en español clasificado.

Ciertamente las opiniones agresivas emitidas en las redes sociales, podrían ser consideradas una forma de ejercer violencia contra las mujeres, toda vez que esté tipo de expresiones en que se evidencie agresividad, desprecio, o menoscabo hacia las mujeres, pueden afectar negativamente en un aspecto psicológico la salud de las víctimas, situación que naturalmente vulnera su derecho a vivir en un entorno seguro y libre de este tipo de apremios y amenazas.

1.1. Violencia contra la mujer en Chile y el mundo.

La violencia contra la mujer es un importante problema de salud pública y una violación de los derechos humanos. (Organización Mundial de la Salud, 2013) Para las mujeres en muchas partes del mundo, la violencia es una de las principales causas de lesiones y discapacidad y un factor de riesgo de sufrir otros problemas de salud física, mental, sexual y reproductiva. La violencia tiene consecuencias a largo plazo para estas mujeres y sus hijos, así como costos sociales y económicos para toda la sociedad. Muchos acuerdos internacionales, incluso la Declaración Universal de los Derechos Humanos de las Naciones Unidas y la Declaración sobre la Eliminación de la Violencia contra la Mujer, han reconocido el derecho humano fundamental de las mujeres a vivir una vida libre de violencia y es por ello que las Naciones Unidas definen

en términos generales la violencia contra la mujer, como todo acto que produce o puede producir un daño a la salud física, sexual o mental de las mujeres.

Puede decirse que sobre Chile pesan obligaciones internacionales de distinta fuente y contenidos. Dichas obligaciones, por ser normas de DDHH, constituyen límites a la soberanía y obligaciones del Estado de rango constitucional. El contenido de esas obligaciones puede ser de respetar los derechos consagrados, protegerlos de la posible interferencia a manos de funcionarios del estado y particulares, así como adecuar la legislación nacional para lograr de forma efectiva la consecución de esos fines.

En materia de VIF son relevantes, a lo menos, cuatro derechos internacionalmente consagrados, el derecho a la libertad, a la igualdad, a la seguridad personal y a una vida libre de tratos crueles, inhumanos o degradantes (Violencia contra la mujer en Chile y derechos humanos, 2017). En virtud del derecho a la libertad, el Estado de Chile debe tomar las medidas necesarias para que las mujeres, al interior de familias en que sufren de violencia, sean protegidas de ella, permitiéndoles efectivamente ordenar sus vidas conforme a sus legítimos intereses.

Una revisión de la normativa vigente relacionada con la violencia contra la mujer dentro de una relación de pareja, da cuenta de algunos cuerpos legales que la regulan tales como: la Ley N° 20.066 sobre Violencia Intrafamiliar, la Ley N° 19.968 que crea los Tribunales de Familia, Código Penal, la Ley N° 20.480 que establece el delito de femicidio y Leyes Especiales, entre ellas, la Ley N° 17.798 sobre Control de Armas y la Ley N° 20.931 “Ley de Agenda Corta”.

Chile todavía no alcanza a cumplir el estándar internacional en cuanto a la regulación del fenómeno. Si bien han existido avances en la protección de la mujer, todavía se ve resguardada mayoritariamente en la intimidad de la relación familiar, reflejando una concepción conservadora de la mujer como persona digna de ser amparada solamente en la lógica de la familia. Se excluye a la mujer de una protección especial cuando se trata de otros tipos de relaciones afectivas tales como los comúnmente denominados “pololeos”, así como cuando se

trata de relaciones entre personas del mismo sexo, sin perjuicio de alguna interpretación jurisprudencial aislada al respecto.

Es menester insistir en que se han realizado esfuerzos por prevenir y erradicar la violencia contra la mujer, tal vez no todos los deseables, pero existen. Ellos provienen principalmente del Ejecutivo, que ha promovido políticas públicas tendientes a crear una institucionalidad para estos efectos, como lo demuestra la instauración primero, del Servicio Nacional de la Mujer y más tarde, del Ministerio de la Mujer y de la Equidad de Género y del Sernam. Así también, ha enviado iniciativas para reformas legales impulsando la labor parlamentaria, la que a su vez se ha visto acrecentada presentando sus propias iniciativas.

Con todo, las políticas públicas en materia de VIF todavía tienen un enfoque fuertemente reactivo e individual frente a la producción de la violencia, olvidando la intervención preventiva y disuasiva. Efectivamente existen campañas generales contra la violencia impulsada tanto por el Estado como por la sociedad civil, no obstante, el círculo de violencia comprende no solo la violencia física sino también la psicológica, y tratándose de esta última, ésta se produce a través de comentarios burlescos, insultos y humillaciones. Es en esta medida que el ciclo de violencia avanza, se producen también amenazas, a veces subestimadas en su seriedad y verosimilitud por parte de los actores judiciales, así como también por las mismas víctimas, que entran de esta forma en un proceso de normalización de hechos violentos. Es por ello que se pretende colaborar con una forma de evaluación primaria y automatizada, que sirva a las víctimas de violencia contra la mujer o sus familiares o amigos cercanos, a contar con una herramienta que les permita detectar en forma oportuna este tipo de conductas que pueden en definitiva agravarse y terminar en una desgracia evitable y tomar medidas al respecto.

1.2. Machine Learning.

Los mensajes publicados en redes sociales como Twitter o Facebook se han transformado en un material de gran interés para detectar tendencias de opinión entre los usuarios. El hecho de

que se hagan públicas opiniones, ideas y debates, da pábulo a asimilarlos con una cierta conversación informal. En el contexto de la comunicación humana, el análisis de contenido y los estudios cuantitativos de los mensajes de Twitter permiten identificar patrones de comportamiento entre los usuarios (Jungherr, 2015).

Sin embargo, la investigación en comunicación requiere complementar los análisis cuantitativos con consideraciones de orden cualitativo. Dado el gran volumen de mensajes de Twitter que habitualmente hay que evaluar, conviene desarrollar métodos que procesen textos de forma automática con una fiabilidad de precisión aceptable. De esta forma, el investigador estaría en condiciones de cualificar mejor las opiniones y los datos extraídos de la conversación entre los usuarios. El análisis de sentimiento en Twitter surge como respuesta a esta necesidad.

El desarrollo de las técnicas de análisis de sentimiento en inglés tiene ya un cierto recorrido con estudios satisfactorios, encontrando entre ellos (Bravo-Marquez, Mendoza, & Poblete, 2014); (Da Silva, Hruschka, & Hruschka, 2014); (Medhat, Hassan, & Korashy, 2014); (Yu & Wang, 2015). Su aplicación al ámbito de la comunicación política en Twitter ha permitido alcanzar conclusiones relevantes para evaluar aspectos relacionados con la influencia (Caton, Hall, & Weinhardt, 2015); (Dang-Xuan, Stieglitz, Wladarsch, & Neuberger, 2013) y con la agenda-setting (Guo & Vargo, 2015). También estas técnicas se han aplicado a otros idiomas, como el portugués (Prata, Soares, Silva, Trevisan, & Letouze, 2016). No obstante, el desarrollo de estas técnicas aplicadas al castellano se encuentra todavía en fase de investigación y desarrollo.

Los sentimientos o emociones son un subtipo de actitudes definidas como relativamente duraderas, creencias afectivas, preferencias y predisposiciones hacia objetos o personas (Scherer, 1984, págs. 37-63). El procesamiento de lenguaje natural (NLP) es una de las técnicas utilizada para identificar estos sentimientos o emociones y es mejor conocida análisis de subjetividad y se puede dividir en Análisis de Sentimiento (SA) y Análisis de Afecto (AA).

Dado el gran volumen de mensajes de Twitter que habitualmente hay que evaluar, conviene desarrollar métodos que procesen textos de forma automática con una fiabilidad de precisión aceptable. De esta forma, el investigador estaría en condiciones de calificar mejor las opiniones y los datos extraídos de la conversación entre los usuarios es por ello que el análisis de sentimiento en Twitter surge como respuesta a esta necesidad.

2. PERFIL DEL PROYECTO

2.1. Objetivo General.

Construir un modelo de detección de agresividad hacia la mujer en opiniones publicadas en redes sociales mediante la aplicación de técnicas de machine learning.

2.2. Objetivos Específicos.

1. Estudiar la literatura y trabajos relacionados con la detección automática de agresividad hacia la mujer, que utilicen técnicas de sentic computing.
2. Recolectar opiniones publicadas en twitter y construir un corpus, que contenga opiniones etiquetadas y que sirva para el entrenamiento y aprendizaje del modelo.
3. Aplicar técnicas de machine learning para la construcción de un clasificador de opiniones agresivas contra la mujer, analizando características utilizadas en la clasificación y las métricas de rendimiento obtenidas.
4. Construir un software que sirva de prototipo y que permita validar los resultados obtenidos del modelo.

2.3. Alcances de la Investigación.

- En este proyecto solamente se clasificará la polaridad de los comentarios, es decir si se aprecia Agresivo o No Agresivo contra las mujeres, sin que se contemple clasificar otras emociones como las propuestas por Plutchik, Ekman, Tomkins o Lazarus, ni tampoco realizar algún granulado con mayores o menores intensidades en los dichos.

- En este trabajo se utilizarán solo algunos clasificadores ya conocidos, no se pretende proponer un nuevo método de clasificación, sino que se espera poder elegir dentro de los ya existentes en la experimentación, los que mejor rendimiento tienen en los análisis para su uso en los procesos de predicción.
- En la presente investigación no se propondrán nuevos parámetros de métrica para analizar el rendimiento de los clasificadores, sino que se evaluará con aquellos ya existentes y que la literatura menciona como los más importantes.
- En este trabajo no se aplicarán técnicas de tratamiento de negación e ironía. Sólo se utilizarán las técnicas de limpieza más adecuadas para el corpus y agregación de features que puedan mejorar el rendimiento de predicción.

2.4. Metodología.

La metodología de trabajo definida para este Proyecto de Título ha sido desarrollada con el fin de cumplir de la mejor forma posible los objetivos establecidos en el numeral 2.2. del presente informe. Las actividades previstas para este proyecto, luego de recopilar la literatura y trabajos relacionados con la detección automática de agresividad hacia la mujer, que utilicen técnicas de sentic computing, continúan con la construcción y etiquetado del corpus, para lo cual es necesario identificar el dominio de trabajo, que en este caso serán las opiniones publicadas en twitter. Junto con ello, será necesario identificar la categoría o polaridad con la cual se clasificarán los comentarios, es decir si los comentarios son **Agresivos** o **No Agresivos** hacia la mujer.

Con el objetivo de mejorar los resultados obtenidos por los clasificadores a estudiar, se aplican ciertas transformaciones al corpus y que tienen relación con la limpieza del set de comentarios etiquetado. Conteste con lo anterior, las transformaciones utilizadas fueron: corrección de errores ortográficos, eliminación de Stop Words, eliminación signos de puntuación, de caracteres o palabras repetidas, entre otras. Asimismo, fue necesario considerar ciertos atributos (features) que permitan hallar patrones dentro de los comentarios para así ayudar a predecir la

clasificación de las opiniones expresadas en un texto. Para esto último, se utilizó la técnica de etiquetado POS.

Como resultado de los procesamientos mencionados anteriormente y aplicados al corpus, se construyeron cuatro versiones distintas de corpus, sobre las cuales se procedió a la aplicación cuatro técnicas de clasificación, que fueron seleccionadas de acuerdo con la conveniencia definida en el desarrollo de las etapas anteriores. Todo ello, con el sentido de analizar la incidencia en el rendimiento de un clasificador al proceder con la aplicación de Stop Words y del etiquetado POS en los corpus generados. Posteriormente los resultados obtenidos por los clasificadores son analizados mediante el cálculo de las métricas Accuracy, Precision, Recall y F-measure.

Finalmente, los resultados obtenidos por los clasificadores serán observados y evaluados de manera individual y en su conjunto, pronunciándose respecto del rendimiento de los clasificadores en las distintas versiones utilizadas del corpus. En la ilustración 1 se expone un resumen con las actividades descritas.



Ilustración 1: Diagrama con Metodología de Trabajo.

3. MARCO TEORICO

3.1. Análisis del Problema de Violencia contra la mujer en Chile.

La violencia es tan antigua como la humanidad misma, como igualmente antiguos son los intentos por limitarla. En ese contexto se inserta la Violencia Contra la Mujer (**VCM**), la que en principio, no debería ser distinta de cualquier otro tipo de violencia, sin embargo, su carácter de elemento estructural dentro de la sociedad la configura como un problema especial. En otras palabras, cuando los actos de violencia ocurren únicamente debido a la condición biológica de la víctima y de las consecuencias sociales culturalmente atribuidas a ella, dichos actos deben ser tratados y combatidos de forma especial. En el plano jurídico, el problema de la VCM debe ser entendido como uno de derechos humanos. Así, la VCM podría ser atacada desde los instrumentos internacionales generales, sin embargo, el carácter de problema estructuralmente perpetuado ya mencionado obliga a la comunidad internacional a valerse de herramientas especialmente pensadas para su erradicación.

En virtud del derecho a la libertad, el Estado de Chile debe tomar las medidas necesarias para que las mujeres, al interior de familias en que sufren de violencia, sean protegidas de ella, permitiéndoles efectivamente ordenar sus vidas conforme a sus legítimos intereses.

La ley N° 20.066 en su artículo 5 señala que se entenderá por violencia, que tipos de violencia comprende y los sujetos que participan, este artículo consagra que: “Será constitutivo de violencia intrafamiliar todo maltrato que afecte la vida o la integridad física o psíquica de quien tenga o haya tenido la calidad de cónyuge del ofensor o una relación de convivencia con él; o sea pariente por consanguinidad o por afinidad en toda la línea recta o en la colateral hasta el tercer grado inclusive, del ofensor o de su cónyuge o de su actual Conviviente. También habrá violencia intrafamiliar cuando la conducta referida en el inciso precedente ocurra entre los padres de un hijo común, o recaiga sobre persona menor de edad, adulto mayor o discapacitada que se encuentre bajo el cuidado o dependencia de cualquiera de los integrantes del grupo familiar” (Congreso Nacional , Chile, 2005).

Ingeniería de Ejecución en Computación e Informática - UBB

Las reformas realizadas a la Ley N° 20.066 son pocas pero sustanciales, siendo relevantes aquellas realizadas por la Ley N° 20.286 de 2008, la Ley N° 20.480 de 2010 y la Ley N° 21.013 de 2017. Por otro lado, se introduce al adulto mayor como sujeto vulnerable para el maltrato y la VIF por la Ley N° 20.427, así como la creación del Ministerio de la Mujer y de la Equidad de Género por la Ley N° 20.820.

El Instituto Nacional de Derechos Humanos (INDH) es una “corporación autónoma de derecho público” creada en el año 2009 por la Ley N° 20405, con el fin de promover y proteger los derechos humanos de todas las personas en territorio chileno. El INDH fue creado con el doble propósito de constituir garantía de no repetición en contexto transicional, y de crear una institución que implementara las obligaciones internacionales garantizando los derechos humanos (INDH, Chile, 2009).

Es así que en 2013 el INDH recomendó al Estado chileno que actuara en pos de la protección de la mujer a través de estrategias que permitan detectar la VCM desde sus tempranas manifestaciones, como por ejemplo en los servicios de salud, evitando que la denuncia constituya la única forma a través de la cual las mujeres víctimas puedan ser auxiliadas.

La violencia contra la mujer es un fenómeno que no ha estado exento de preocupación por parte de la comunidad internacional, que ha ido dictando normas y creando estándares, primero con la Convención sobre la Eliminación de todas las formas de Discriminación contra la Mujer, y posteriormente la Declaración y Plataforma de Beijing. En el ámbito regional, destaca la Convención Interamericana para Prevenir, Sancionar y Erradicar la Violencia contra la mujer (“Convención de Belém do Pará”). Todos estos instrumentos tuvieron por objeto el proteger a la mujer en su vulnerabilidad, pero también empoderarla en cuanto sujeta activa que puede, y debe, ejercer sus derechos políticos, civiles, sociales y culturales.

Actualmente, y desde 2005, existen los Centros para la Mujer, en donde se atiende sólo a las mujeres víctimas (sin perjuicio de ofrecer orientación también a quienes son testigos de la violencia que sufren las mujeres). En forma paralela, en 2009 se crearon los Centros de Tratamiento de Hombres que Ejercen Violencia de Pareja (Centros HEVPA), pero ellos son insuficientes para cubrir la demanda y proporcionar tratamientos que anticipen el acontecimiento de hechos violentos tanto por parte de agresores actuales como de futuros (por ejemplo, hijos). No hay entonces, desde la institucionalidad, una mirada estructural de la problemática, que actúe de forma estratégica y no solo una vez que haya víctimas (Violencia contra la mujer en Chile y derechos humanos, 2017).

Finalmente se señala que se ha avanzado como país en el tratamiento de la VCM desde la dictación de la primera Ley de VIF de 1994 a la actualidad, pero el enfoque está dirigido al contexto intrafamiliar. Hoy existe una institucionalidad con el objetivo de proteger a la mujer frente a la violencia, y se han hecho reformas para mejorar la legislación y agilizar los procedimientos en los tribunales. Pero también en nuestra sociedad han sido muchos los actores que tratan de hacer aportes en la prevención y tratamiento de este flagelo, siendo ésta la principal motivación de este proyecto, cuyo objetivo es proponer una herramienta que pueda ser utilizada en la detección oportuna de conductas que evidencien una actitud agresiva hacia cualquier mujer, para que éstas puedan ser a tiempo consideradas y se realicen acciones que tiendan evitar consecuencias mayores.

3.2. Machine Learning.

El Análisis de Sentimiento (AS o SA por sus siglas del inglés *Sentiment Analysis*) es un campo de investigación dentro del PLN que intenta extraer de forma automática y a través de técnicas computacionales, información subjetiva expresada en el texto de un documento dado y acerca de un determinado tema. De esta forma, mediante el análisis de sentimiento podremos determinar si un texto presenta connotaciones positivas o negativas. Entre varios autores que

han definido el análisis de sentimiento, es la definición de (Cambria & Hussain, 2012) la que más se ajusta a nuestro interés.

“Conjunto de técnicas computacionales para la extracción, clasificación, comprensión y evaluación de opiniones expresadas en fuentes publicadas en Internet, comentarios en portales web y en otros contenidos generados por usuarios.”

Se puede apreciar que sólo hace referencia a las opiniones, dejando fuera del alcance de estudio a los sentimientos y a la subjetividad. Aunque es probable que dejar fuera a los sentimientos sea un error puesto que muchas veces las opiniones están fundamentadas y emanan de los sentimientos de quien las expresa, pero como indica E. Martínez en (Martínez Camara, 2015), sí es un acierto no hacer referencia a la subjetividad ya que las opiniones se pueden encontrar en oraciones subjetivas y también objetivas.

La polaridad de los sentimientos es una característica particular de los comentarios. Esta se hace presente regularmente de forma dicotómica, es decir positivo o negativo, a pesar de que también puede ser vista dentro de un rango. Un documento posee varias frases que demuestran opiniones, las cuales podrían tener una polaridad mixta, que es diferente a que estas no tuviesen polaridad, yendo más lejos, se debe hacer una distinción entre la polaridad del sentimiento y la fuerza que este tiene.

El proceso entonces de clasificación pasa a ser dicotómico, es decir solo tomará dos puntos, positivo y negativo. El texto en general debe pertenecer a una de las dos categorías, en las investigaciones realizadas previamente por algunos autores (Mejova, 2009) el nivel de dificultad disminuye al procesar, por ejemplo, críticas de productos, ya que poseen bien definidos el sentido que se da en el texto, sin embargo, existen otros casos en donde no es tan sencillo la clasificación de la polaridad, como es el caso de las noticias, ya que dentro del texto pueden aparecer una serie de sentimientos y el discriminar que es “bueno” o “malo” se torna más difícil de discernir.

Para llevar a cabo la clasificación de un documento en base a su sentimiento existen diversos métodos y técnicas que se van refinando y mejorando a medida que avanzan las investigaciones sobre esta materia y aparecen en escena nuevos estudios y trabajos. A pesar de la multitud de artículos y publicaciones presentados cada año, cuestión que pone de manifiesto una corriente de investigación en pleno apogeo y crecimiento, no parece existir un consenso claro sobre qué técnicas se deben usar para obtener los mejores resultados en el proceso de clasificación de textos. Y es debido a este gran número de publicaciones y a un campo de investigación sumido en un proceso de fuerte expansión por lo que no es sencillo fijar una división clara de los métodos existentes en la actualidad. Aun así, varios autores como (Liu, 2012) o (Biagioni, 2016) establecen dos grandes grupos, métodos supervisados y no supervisados.

3.3. Clasificación de Métodos de Machine Learning.

3.3.1. Mediante aprendizaje supervisado.

La clasificación mediante técnicas de aprendizaje supervisado está basada en el uso de algoritmos de aprendizaje automático, conocidos también como *machine learning*. Su tipificación de “supervisados” obedece al hecho de que necesitan de un grupo de documentos de ejemplo previamente etiquetados para generar un modelo que será usado posteriormente para clasificar nuevos textos y que en este contexto es conocido como **Corpus**. Su funcionamiento se basa en la relación matemática creada entre los elementos de ejemplo durante un proceso conocido como entrenamiento y en donde se genera un modelo estadístico que agrupa dichos elementos en tantos conjuntos como diferentes etiquetas o clases existan en el grupo de documentos de entrenamiento. Posteriormente, el modelo generado se utiliza con un ejemplo no etiquetado para determinar de cuál de los grupos existentes formaría parte, realizando así una predicción en base a los ejemplos aportados durante la fase de entrenamiento.

El éxito y efectividad de los sistemas de aprendizaje automático a la hora de clasificar nuevos elementos depende principalmente de dos factores: del algoritmo de clasificación seleccionado y de las características o *features* elegidas para representar los elementos de ejemplo y con los que entrenar dicho algoritmo, no obstante, también se debe hacer hincapié en la importancia del nivel de acuerdo que resulte del proceso de etiquetado humano del corpus. Existen variados algoritmos de aprendizaje automático diferentes cuyos resultados además pueden ser mejorados mediante la configuración de sus diferentes parámetros; pero son las características elegidas para el entrenamiento las verdaderamente importantes y de ellas depende en gran medida el éxito de este tipo de métodos de clasificación.

3.3.2. Mediante aprendizaje no supervisado.

Para resolver el hecho de tener que contar con ejemplos etiquetados y salvar el problema de la dependencia de dominio, aparecen en escena los sistemas de clasificación mediante aprendizaje no supervisado. Estos sistemas tratan de inferir la polaridad del sentimiento global de un documento a partir de la orientación semántica de las palabras o frases que lo conforman. Existen dos enfoques para la resolución del problema de clasificación de textos mediante aprendizaje no supervisado: métodos basados en diccionarios y métodos basados en relaciones lingüísticas.

3.4. Preparación de los datos: limpieza del corpus.

En todo método que se haga uso de algoritmos de aprendizaje automático es necesario tratar previamente los datos con los que serán entrenados. El objetivo de esta fase es limpiar y normalizar la información para impedir que ciertos datos puedan influir de manera negativa en el resultado final. Lo anterior es decisivo cuando hablamos de mensajes extraídos de redes sociales ya que es muy habitual encontrar mensajes con faltas de ortografía, repeticiones de caracteres, mezcla de letras mayúsculas y minúsculas o, en el caso concreto de redes de

microblogging como Twitter, uso de jerga y abreviaturas para escribir mayor contenido en un número tan reducido de caracteres.

Con el fin señalado anteriormente existe un conjunto de reglas sencillas de aplicar y que suelen ser comunes en la construcción de este tipo de clasificadores (Sobrinho, 2018). El objetivo que persiguen todos ellos es la normalización de los mensajes, pero evitando en todo momento que los cambios aplicados provoquen la pérdida de la polaridad de sentimiento. Estas son las reglas a utilizar generalmente:

Normalización de mayúsculas y minúsculas: aunque para las personas es sencillo saber que las palabras “barco” y “BARCO” tienen exactamente el mismo significado, para los algoritmos de aprendizaje automático esto no así. De hecho, son tratadas como palabras totalmente distintas, sin ningún tipo de relación entre ellas. Para evitar que esto suceda y mantener el significado de las palabras sin tener en cuenta la forma de sus caracteres, todos los mensajes serán convertidos a su equivalente en letras minúsculas.

Tratamiento de la duplicidad de caracteres: en las redes sociales es habitual repetir las mismas letras en las palabras para dar intensidad a lo que se intenta expresar. Por ejemplo, no es lo mismo escribir “Qué calor hace” o “Qué calooooooooo haceeeeeee”. Aunque ambas frases conceptualmente son equivalentes, la segunda enfatiza la sensación de calor mediante la repetición de los caracteres de la frase.

Eliminación de tildes: en las redes sociales los usuarios no acostumbran a hacer un buen uso de las tildes. Por esta razón, las palabras “alegría” y “alegria” (sin tilde en la letra i) serían consideradas por los algoritmos como distintas. Para evitar esta pérdida de relación semántica, serán eliminadas todas las tildes de las vocales de los mensajes de entrenamiento.

Eliminación de números: por norma general, las cifras numéricas no suelen contener información que ayude al proceso de clasificación de polaridad de sentimiento por lo que serían removidas de los textos y así ayudar a reducir la cantidad de características del corpus.

Eliminación de *retweets*: aunque en la actualidad esto ya no es así, hace unos años, justo cuando se extrajeron gran parte de los tweets que forman el corpus de entrenamiento, las menciones a las publicaciones de otros usuarios se hacían mediante el uso de la palabra reservada “RT” seguida del nombre de usuario y del texto a referenciar. Esta palabra clave no aporta información relevante para el proceso de clasificación por lo que será eliminada de todos los mensajes.

Eliminación de retornos de carro: algunos mensajes de Twitter contienen saltos de línea y retornos de carro, por lo que el texto aparece escrito en diferentes líneas. Estos elementos serán también eliminados para que los mensajes aparezcan escritos en una sola línea.

Normalización de jerga: en las redes sociales se utiliza un lenguaje coloquial y muy informal que hace uso intensivo de abreviaturas y secuencias de caracteres sin aparente sentido. Esta jerga aumenta en Twitter debido a la limitación del número de caracteres permitidos por mensajes. Así, es habitual escribir “q” en lugar de “que”, “tb” en lugar de “también” o el clásico “tqm” para referirse a “te quiero mucho”. El proceso de normalización tendrá en cuenta esta realidad y aplicará un conjunto de reglas para sustituir estas abreviaturas por sus palabras reales.

3.5. Tokenización.

Una vez completado el proceso de normalización de los mensajes del corpus, la siguiente etapa es la denominada *tokenización*. En esta fase los textos se dividen en unidades más pequeñas llamadas *tokens* y que normalmente se corresponden con las palabras de cada texto. Este proceso puede ser tan sencillo como separar los términos de las frases por los espacios en blanco y los caracteres de puntuación o bien considerar además que la agrupación de determinados

símbolos puede contener algún tipo de información que sea útil al proceso de clasificación. Este podría ser el caso de los emoticonos, secuencias de caracteres de puntuación que suelen ser un indicador de la polaridad del sentimiento de las palabras a las que acompañan.

3.6. Reducción de las características.

Esta fase es opcional y su objetivo es disminuir el número de características del corpus mediante la eliminación de determinados *tokens* o de su conversión buscando una misma manera de representarlos. Existe tres técnicas habituales para llevar a cabo esta tarea: eliminación de *stopwords*, lematización y *stemming*.

Eliminación de stopwords: existe un conjunto de palabras que, aunque son necesarias para construir oraciones con sentido, carecen de información que ayude a determinar la polaridad de los textos en los que se encuentran. En español, estas palabras son las preposiciones, los pronombres, las conjunciones y las distintas formas del verbo haber, entre otras. Mediante esta técnica, todos los términos pertenecientes a la lista de *stopwords* serán eliminadas del modelo antes del entrenamiento de los algoritmos.

Lematización: este es un proceso de normalización morfológica que transforma cada palabra en su lema mediante el uso de diccionarios y de un proceso de análisis morfológico. A modo de ejemplo, la lematización convertiría la palabra “guapas” a su lema “guapo”. Por tanto, muchas características tomarían la misma forma, reduciendo así su variabilidad.

Stemming: se trata de otro método de normalización morfológica pero más agresivo que la lematización. En este caso, una palabra se transforma a su raíz por medio de la supresión de sus sufijos e inflexiones. Siguiendo el ejemplo anterior, la palabra “guapas” se convertiría a su raíz, “guap”.

3.7. Features etiquetado POS.

El etiquetado gramatical (part-of-speech tagging, POS tagging o POST) es el proceso que recibe como entrada texto en algún lenguaje y como salida regresa un conjunto de pares de la forma palabra-etiqueta gramatical, basado en su definición y su contexto, categorías gramaticales como sustantivo, verbo, adjetivo, etc. Las categorías gramaticales resultan de gran utilidad por la gran cantidad de información que dan acerca de una palabra y sus vecinos. Saber si una palabra es un sustantivo o un verbo nos dice mucho sobre las palabras junto a ellas y las formas de interpretarlas y sirve para encontrar entidades nombradas, es decir, nombres de ciudades, negocios, objetos, personas o cualquier otra cosa que pueda llevar un nombre, en textos y en otras tareas de extracción de información.

El etiquetado gramatical resulta difícil porque una misma palabra puede representar dos o más categorías diferentes. Si se tienen los siguientes dos ejemplos de oraciones: “Yo he hecho mis deberes” y “El más reciente hecho provocó los cambios”, se puede ver que la palabra “hecho” se refiere a dos categorías distintas en cada oración. Realizar correctamente el etiquetado implica marcar la palabra como un verbo en la primera oración y como sustantivo en la segunda. Adicionalmente, en la práctica se suelen usar muchas más categorías, esto para hacer referencia a propiedades de la palabra; en el primer ejemplo “hecho” sería “verbo en participio” y en el segundo “sustantivo masculino”. Una vez que se tienen las categorías gramaticales, se pueden tomar diferentes acciones para una misma palabra dependiendo de la categoría asignada.

3.8. Stop words.

Stop words es el nombre que reciben las palabras sin significado como artículos, pronombres, preposiciones, etc. que son filtradas antes o después del procesamiento de datos en lenguaje natural (texto). Para este proyecto, se dispone de un conjunto de stop words de 352 palabras, las cuales son el resultado de una búsqueda y unión de varios diccionarios de este tipo de palabras existentes en la red. Se tienen por ejemplo dentro de este conjunto: algún, alguna,

algunas, alguno, algunos, ambos, ante, antes, etc. Para la carga de estas, normalmente se tiene una función, la cual carga el txt de stop words y retorna un array con este cargado, la cual será instanciada cada vez que el programa lo requiera para limpiar las palabras.

3.9. Software WEKA

Weka (Waikato Environment for Knowledge Analysis, en español “entorno para análisis del conocimiento de la Universidad de Waikato”) es una plataforma de software para el aprendizaje automático y de minería de datos escrito en Java y desarrollado en la Universidad de Waikato en Nueva Zelanda. Weka es un software libre distribuido bajo la licencia GNU-GPL (Hall, y otros, 2009).

Contiene una colección de herramientas de visualización y algoritmos para análisis de datos y modelado predictivo, unidos a una interfaz gráfica de usuario para acceder fácilmente a sus funcionalidades. La versión original de Weka fue un front-end en TCL/TK para modelar algoritmos implementados en otros lenguajes de programación, más unas utilidades para pre-procesamiento de datos desarrolladas en C para hacer experimentos de aprendizaje automático. Esta versión original se diseñó inicialmente como herramienta para analizar datos procedentes del dominio de la agricultura, pero la versión más reciente basada en Java (WEKA 3), que empezó a desarrollarse en 1997, se utiliza en muchas y muy diferentes áreas, en particular con finalidades docentes y de investigación.

Dentro de los aspectos más relevantes de Weka, destacan: licencia pública general de GNU, portabilidad; está completamente implementado en Java y puede correr en casi cualquier plataforma, contiene una extensa colección de técnicas para pre-procesamiento de datos y modelado, es fácil de utilizar gracias a su interfaz gráfica de usuario. Weka soporta varias tareas estándar de minería de datos, especialmente, pre-procesamiento de datos, clustering, clasificación, regresión, visualización, y selección. Todas las técnicas de Weka se fundamentan en la asunción de que los datos están disponibles en un archivo plano (flat file) o una relación,

en la que cada registro de datos está descrito por un número fijo de atributos (normalmente numéricos o nominales, aunque también se soportan otros tipos), también proporciona acceso a bases de datos vía SQL gracias a la conexión JDBC (Java Database Connectivity) y puede procesar el resultado devuelto por una consulta hecha a la base de datos.

WEKA tiene integrados filtros que permiten realizar manipulaciones sobre los datos en dos niveles: atributos e instancias. Las operaciones de filtrado pueden aplicarse “en cascada”, de manera que cada filtro toma como entrada el conjunto de datos resultante de haber aplicado un filtro anterior. En relación a la clasificación, Weka incorpora diversos algoritmos de clasificación automática, dentro de los que se incluyen los mencionados en apartados anteriores. Estos algoritmos son utilizados sobre un set de datos y realizan la clasificación mediante métodos de set de entrenamiento o de validación cruzada:

- **Hold-out (Set de entrenamiento):** Es el método más sencillo de evaluación y consiste en dividir el conjunto de datos en dos partes: entrenamiento y prueba (generalmente se utilizan dos tercios para entrenamiento y un tercio para prueba). Utilizando este método se obtienen resultados para el conjunto de datos de prueba elegido, pero se ignora el comportamiento del clasificador para otros conjuntos de prueba, por lo que podría ocurrir que el clasificador funcione bien para los documentos de prueba elegidos, pero no en otros casos.
- **K-fold cross-validation (Validación cruzada):** Este método es una mejora del método Hold-out. En este método el conjunto de datos se divide en k partes, cada una denominada fold, y se repite el método Hold-out k veces utilizando un fold como conjunto de prueba y el resto (k-1 fold) como conjunto de entrenamiento. Luego, las métricas del clasificador se calcularán como el promedio de las mediciones obtenidas en cada ejecución.

3.10. Técnicas y Algoritmos.

Los algoritmos de aprendizaje supervisado se pueden dividir principalmente en dos grandes grupos: de regresión y de clasificación. Los primeros permiten inferir un valor numérico a partir de una serie de datos de entrada, por ejemplo, las ventas que tendrá una determinada empresa. En cambio, los de clasificación se utilizan para deducir a qué grupo pertenece un ejemplo dado de entre los grupos disponibles. Aunque ambos tipos de algoritmos pueden ser usados en el análisis de sentimientos, en nuestro proyecto usaremos cuatro algoritmos de clasificación muy populares y que ya han sido utilizados en múltiples ocasiones para esta tarea: Naive Bayes, máquinas de vectores de soporte, random forest y árboles de decisión.

3.10.1. Clasificador de Naive Bayes.

La familia de algoritmos Naive Bayes están basados en el célebre Teorema de Bayes, el cual dice lo siguiente:

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

$$P(A_i|B) = \frac{P(B|A_i) * P(A_i)}{P(B)}$$

Ecuación 1: Clasificador Naive Bayes

En el caso concreto de la clasificación de textos, los sucesos excluyentes y exhaustivos son las diferentes clases que se pueden asignar a un mensaje, de manera que no es posible asignar más de una simultáneamente (excluyentes) y esas clases son todos los tipos que existen

(exhaustivos). Los algoritmos Naive Bayes suelen recibir el apelativo de “ingenuos” debido a que en sus cálculos las características seleccionadas para representar a los ejemplos de entrenamiento son estadísticamente independientes y contribuyen por igual en el proceso de clasificación. Dicho de otro modo y en el caso concreto de la clasificación de textos, se considera que las palabras de un mismo mensaje no mantienen ningún tipo de relación entre sí y es indiferente la posición que tienen dentro del texto al que pertenecen.

3.10.2. Clasificador Árboles de Decisión (Decision Tree - Algoritmo C4.5 (J48)).

El algoritmo C4.5 construye árboles de decisión de un sistema de datos del entrenamiento de la misma forma que el algoritmo ID3, que usa el concepto de entropía de la información. Los datos del entrenamiento son un sistema $S = s_1, s_2, \dots$ de muestras ya clasificados. Cada ejemplo $s_i = \{x_1, x_2, \dots\}$ es un vector donde x_1, x_2, \dots representan los atributos o características del ejemplo. Los datos de entrenamiento son aumentados con un vector $C = \{c_1, c_2, \dots\}$ donde c_1, c_2, \dots representan la clase a la que pertenece cada muestra. Los árboles de decisión generados por C4.5 pueden ser usados para clasificación, y por esta razón, C4.5 está casi siempre referido como un clasificador estadístico. (Quinlan, 1993).

Los árboles de decisión forman uno de los grupos de algoritmos más reconocidos y utilizados dentro del campo de la Inteligencia Artificial y del aprendizaje automático. Su estructura es la de un grafo dirigido en forma de árbol compuesto por un conjunto de reglas extraídas a partir de las características de los datos de entrenamiento y que se aplican de manera sucesiva a la hora de predecir a qué clase pertenece un nuevo ejemplo. En general, un árbol de decisión está formado por nodos y líneas que unen dichos nodos, comenzando en uno raíz y terminando en varios con las posibles clasificaciones que se pueden establecer a una muestra dada. Partiendo de la raíz, el paso entre los distintos nodos del árbol se lleva a cabo mediante la evaluación de algún tipo de condición y que determina el recorrido que seguirá la muestra hasta encontrar la clase que le corresponde en un nodo terminal (Ver esquema en la ilustración 2).

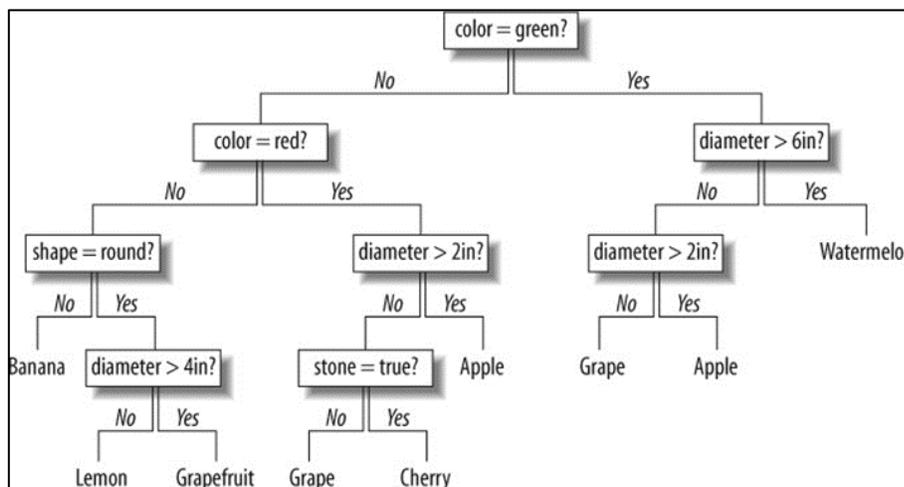


Ilustración 2: Esquema de Árbol de Decisión (Decision Tree).

3.10.3. Clasificador Random Forest.

Random Forest es una combinación de árboles predictivos (clasificadores débiles); es decir, una modificación del Bagging, el cual trabaja con una colección de árboles incorrelacionados y los promedia (Hastie, Friedman, & Tibshirani, 2001), en el cual se tiene que cada árbol depende de los valores de un vector aleatorio de la muestra de manera independiente y con la misma distribución de todos los árboles en el bosque. La generalización de error para los bosques converge a un límite en cuanto el número de árboles en el bosque sea grande.

El elemento común en todos estos procedimientos es que para el k-ésimo árbol se genera un vector aleatorio Θ_k , independiente de los últimos vectores aleatorios $\Theta_1, \dots, \Theta_{k-1}$ pero con la misma distribución; y un árbol se desarrolla usando el conjunto de entrenamiento y de Θ_k , lo que resulta en un clasificador donde $h(x, \Theta_k)$ es un vector de entrada (Breiman, 1999).

Como se ha señalado líneas arriba, el método Random Forest se basa en un conjunto de árboles de decisión, es decir, una muestra entra al árbol y es sometida a una serie de test binarios en cada nodo, llamados split, hasta llegar a una hoja en la que se encuentra la respuesta. Esta

técnica puede ser utilizada para dividir un problema complejo en un conjunto de problemas simples (Ver esquema en la ilustración 3).

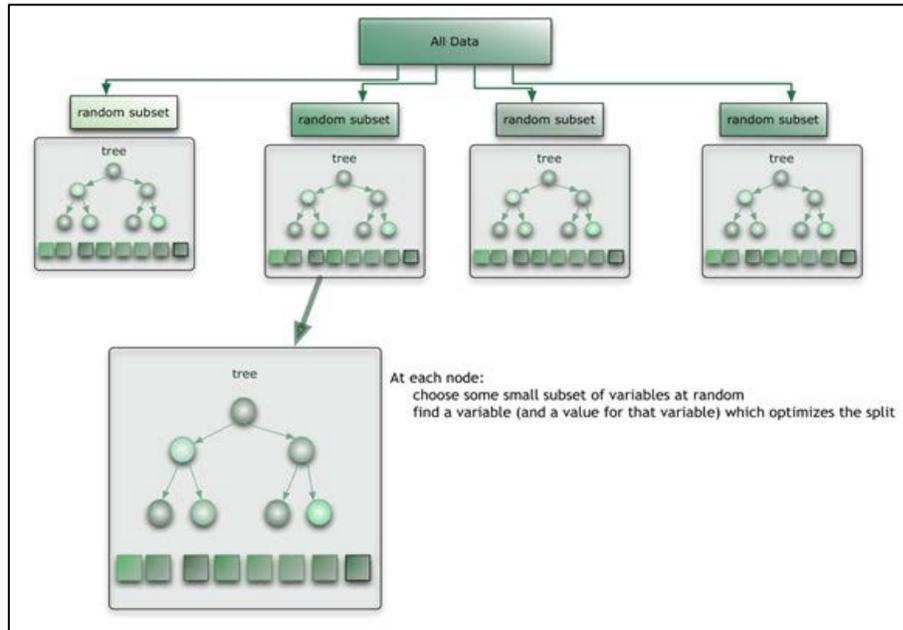


Ilustración 3: Esquema de Bosque Aleatorio (Random Forest).

3.10.4. Clasificador Máquinas de Vectores de Soporte (SVM).

Las máquinas de vectores de soporte (del inglés, Support Vector Machine o SVM) son un grupo de algoritmos de aprendizaje supervisado desarrollados por (Vapnik, 1982) en los laboratorios AT&T. De manera visual, podemos pensar en este tipo de algoritmos como la representación gráfica de un espacio multidimensional en donde se sitúan los puntos que simbolizan los ejemplos de entrenamiento. Un *hiperplano*, denominado vector de soporte, los separa la mayor distancia posible en base a su clase. De esta forma, el vector determina la frontera que sirve para clasificar un nuevo elemento, por lo que dependiendo a qué parte del espacio pertenezca, se le asignará una clase u otra.

Estos algoritmos cuentan con una serie de parámetros que permiten ajustar su configuración interna y así optimizar los resultados durante el proceso de clasificación. Uno de estos parámetros es el *kernel* y se utiliza cuando no es posible separar las muestras mediante una línea recta, plano o hiperplano de N dimensiones, permitiendo tal separación mediante otro tipo de funciones matemáticas como polinomios, funciones de base radial Gaussiana, Sigmoid u otras. Otro de estos parámetros es *regularization* que permite crear un margen blando de manera que se consientan ciertos errores en la clasificación y se evite el sobre-entrenamiento, para terminar, el parámetro *gamma* determina la distancia máxima a partir de la cual una muestra pierde su influencia en la configuración del vector de soporte, y *margin*, que es la separación entre el vector y las muestras de cada clase más cercanas al mismo (Ver esquema en la ilustración 4).

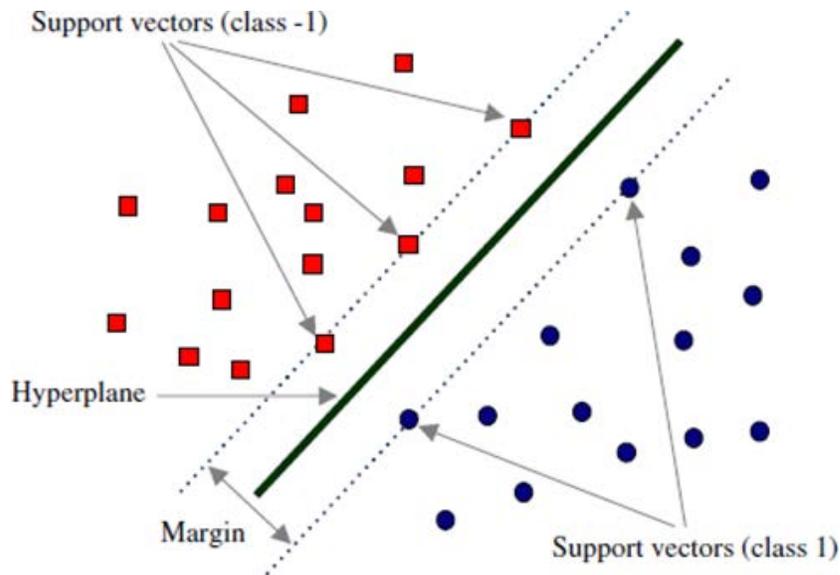


Ilustración 4: Esquema de Máquina de Soporte de Vectores (SVM).

3.11. Métricas y métodos de evaluación de resultados.

Para determinar el rendimiento de los algoritmos y de su configuración, es necesario contar con una serie de medidas que permitan evaluar de manera objetiva su eficacia a la hora de clasificar

los ejemplos que se le proporcionen. Para ello, es importante no tener sólo en cuenta las muestras clasificadas correcta e incorrectamente, sino también las que habiéndose clasificado de manera errónea podrían haberse etiquetado bien. Para entender los cuatro posibles estados de un ejemplo a clasificar, pensemos en una clase A y en un algoritmo que determina si dicho ejemplo pertenece o no a esa clase:

- **True Positives (Verdaderos Positivos o TP):** son los ejemplos que han sido marcados de manera correcta como pertenecientes a la clase A.
- **False Positives (Falsos Positivos o FP):** serán los ejemplos marcados como de clase A, pero en realidad no pertenecen a ella, es decir, han sido clasificados de manera incorrecta.
- **True Negatives (Verdaderos Negativos o TN):** en este caso, los ejemplos no son de la clase A y han sido clasificados correctamente.
- **False Negatives (Falsos Negativos o FN):** en este grupo estarán los ejemplos marcados como no pertenecientes a la clase A, pero en realidad sí lo son y, por tanto, no se han clasificado correctamente.

Teniendo en cuenta los estados anteriores, podemos definir las siguientes medidas que serán usadas para evaluar nuestros modelos:

Exactitud (del inglés Accuracy): esta es la medida de rendimiento más simple e intuitiva y representa la razón entre las predicciones correctas sobre el total de predicciones realizadas. Dicho de otra manera, es el número de elementos clasificados correctamente entre el número total de clasificaciones llevadas a cabo.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Ecuación 2: Métrica Exactitud (Accuracy).

Es común pensar que el modelo que ofrezca una mayor exactitud es el mejor modelo. En realidad, esta medida es adecuada en el caso de que el número de elementos de cada clase sea aproximadamente el mismo y el corpus esté balanceado. En caso contrario, es necesario hacer uso de otro tipo de medidas como la precisión, la exhaustividad y el valor-F. Al contrario de la exactitud, estas medidas no valoran el rendimiento del modelo teniendo en cuenta todas las clases del sistema, sino que lo hacen sobre clases individuales. Dicho de otro modo, la precisión, la exhaustividad y el valor-F ofrecerán valores distintos para la clase A y para la B.

Precisión (del inglés Precision): es la razón entre el número de documentos clasificados correctamente como pertenecientes a la clase A y el número total de documentos de que han sido clasificados por el modelo como de clase A.

$$Precision = \frac{TP}{TP + FP}$$

Ecuación 3: Métrica Precisión (Precision).

La precisión mide la proporción de identificaciones positivas que son realmente correctas. Nótese que su valor aumenta a medida que el número de falsos positivos disminuye.

Exhaustividad (del inglés Recall): es la relación entre los documentos clasificados correctamente como pertenecientes a la clase A y la suma de todos los documentos de la clase A.

$$Recall = \frac{TP}{TP + FN}$$

Ecuación 4: Métrica Exhaustividad (Recall).

La cobertura es la proporción de elementos positivos reales identificados acertadamente. También se puede ver como la capacidad que tiene el modelo de construir de manera correcta las clases. Cuanto más cercano a 1, mejor estarán definidas las distintas clases existentes ya que su valor aumenta a medida que disminuye el número de falsos negativos.

Valor-F (del inglés F-score o F-measure): es habitual que para medir la eficiencia de un modelo de clasificación se haga uso de los valores de cobertura y exhaustividad. Para ello, el valor-F se presenta como la media armónica entre ambas medidas y suele utilizarse como referencia para comparar el rendimiento entre varios modelos. La fórmula del valor-F combina las dos medidas anteriores de manera ponderada a través de un parámetro β lo que permite otorgar una mayor importancia a una que a otra:

$$F_{\beta} = (1 + \beta^2) * \frac{Precision * Recall}{(\beta^2 * Precision) + Recall}$$

Ecuación 5: Métrica Valor-F (F-Measure).

Es frecuente que la precisión y la exhaustividad tengan el mismo peso en la fórmula, es decir, con un valor β igual a 1. A esta configuración se le conoce como Valor-F1 o F1-score.

En el caso de un sistema que cuente con más de dos clases, como el nuestro, se debe calcular cada una de las métricas anteriores para cada clase y combinarlas entre ellas para obtener una medida global. Estas medidas ponderadas serán las que determinarán cuál de nuestros modelos tiene el mejor rendimiento.

3.12. K-fold Cross Validation.

Por otra parte, es importante explicar que los modelos de este apartado serán construidos mediante un sistema de validación cruzada (del inglés cross-validation). Este método reduce la dependencia entre la partición de datos usada para la fase de entrenamiento y para la de pruebas.

La validación cruzada divide el número de muestras del corpus en k conjuntos de igual tamaño de manera que $k-1$ grupos son usados para entrenar el sistema y el grupo sobrante para su evaluación. Este proceso se repetirá k veces y en cada uno de ellos se escogerá un grupo diferente para validar la eficacia del modelo. En cada iteración se calculará el valor-F1 ponderado y se hallará la media entre todos, medida que servirá como referencia de su efectividad. La validación cruzada suele tomar los valores 3, 5 o 10 para el parámetro k . En nuestro caso concreto será de 10.

3.13. Proyectos Relacionados.

En la región podemos señalar que en el vecino país Argentina en el 2018 se lanzó una aplicación para el celular que se llama "**la 144**" y presenta varias opciones donde se obtiene información sobre el tema y lugares donde acudir por asesoría, tiene una encuesta para que cada persona pueda darse cuenta o evaluar si está en una relación violenta o alguien está ejerciendo violencia contra ella. También se encontró otra conocida como **El Botón de Pánico #NiUnaMenos** es una aplicación que está pensada para que funcione en un entorno familiar. A través de la 'app', la víctima tiene la posibilidad de alertar a contactos de confianza predefinidos que puedan intervenir ante casos de emergencia, dicha alerta se envía vía SMS, indicando la situación de emergencia y la ubicación donde se encuentra la persona.

En Chile no existen aplicaciones y todas las gestiones realizadas por el gobierno y otras organizaciones se refieren a centros de atención y campañas educativas para prevenir la violencia de género.

En países de Europa existen más iniciativas al respecto, pudiéndose destacar las siguientes apps:

Libres: Esta app ha sido desarrollada por el Ministerio de Sanidad, Servicios Sociales e Igualdad de España, se trata de una vía para estar en contacto con las mujeres que sufren

violencia de género, a través de la aplicación se les ofrece las herramientas e información necesaria que les permitan acabar con esta situación.

SMS ¡Actualízate! Amor 3.0: La educación es clave para erradicar la violencia de género, por eso el Gobierno de Canarias desarrolló esta app con la finalidad de sensibilizar a los jóvenes a través de tests, interpretación de resultados y recursos. Se trata de una app interactiva que permite aclarar ciertos mitos, concienciar sobre el sexismo y detectar indicios de abusos en una relación. Se trata de una de las más valoradas por los usuarios.

Stop Agresiones: En caso de estar en peligro de agresión, con un simple movimiento de muñeca o un toque en la pantalla, la app empieza a actuar. Stop Agresiones requiere que el GPS del móvil esté activado para poder enviar a través de WhatsApp a los contactos que se han seleccionado previamente una alarma, de esta manera puedes pedir auxilio y enviar tu ubicación exacta.

Relación Sana: Una aplicación orientada de nuevo a la prevención y la detección de la violencia de género entre la juventud. La app, desarrollada por la Consejería de Presidencia del Gobierno Autónomo de Murcia, ayuda a detectar la violencia de género en las relaciones entre adolescentes y fomenta la igualdad en la población juvenil a través de diversos cuestionarios.

Alert Cops: El Ministerio del Interior de España ha creado esta aplicación basada en el servicio de alerta de las Fuerzas y Cuerpos de Seguridad del Estado. A través de la app se puede enviar una alerta, empleando el sistema de geolocalización del teléfono, a la policía que más cerca se encuentre de tu ubicación.

Enrédate sin machismo: El Cabildo de Tenerife sacó esta aplicación contra la violencia machista, la cual se ha convertido en una de las mejor valoradas (4,5 sobre 5). ¿Su objetivo? Ayudar a la población, especialmente a los jóvenes, a detectar indicios de violencia (de cualquier tipo) en las relaciones, a través de diferentes cuestionarios y niveles que hay que

superar se llega a un resultado, si el resultado detecta signos de violencia de género, la app ofrece una guía para resolver la situación.

No obstante, todo lo anterior, ninguna de las aplicaciones mencionadas ofrece las prestaciones de un análisis predictivo, contemplando algún tipo de proceso de análisis de sentimiento o minería de opinión, que sería el método que nos interesaría desarrollar considerándose el vertiginoso desarrollo que ha venido sosteniéndose en los últimos años.

Dentro de los objetivos del presente proyecto de título, se encontraba puntualmente el hecho de utilizar como método de detección de agresividad el proceso el análisis de sentimiento, pero en el medio nacional los escasos trabajos encontrados han sido desarrollados mediante el análisis de afecto, como es el caso del análisis automático de opiniones acerca de comentarios de libros (Oyarzún, 2014) o el desarrollo de la investigación acerca de la comparación de rendimiento de técnicas de aprendizaje automático para análisis de afecto sobre textos en español (Elgueta, 2017), ambos trabajos en la Universidad del Bio Bio, Adicionalmente se encontraron algunos otros proyectos todos de alumnos de la Universidad de Chile (Hernández, 2016), (Córdova , 2015), (Ponce de León, 2015), (Montecinos, 2014), todas los cuales apuntan al análisis de opinión a través de la creación de diccionarios lexicón que permiten la clasificación de nuevos comentarios de forma automática.

Independiente del escaso desarrollo de este tipo de técnicas a nivel nacional, es sumamente importante destacar que el análisis de sentimientos es una de las áreas de investigación de más rápido crecimiento en ciencias de la computación, cabe señalar que el brote del análisis de sentimientos basado en computadora solo ocurrió con la disponibilidad de textos subjetivos en la Web, por lo tanto, el 99% de los artículos a nivel mundial se publicaron después de 2004. Incluso este crecimiento exponencial ha generado trabajos que se refieren simplemente a orientar acerca de los innumerables trabajos que se han realizado en distintas líneas de investigación sobre el particular. Según el estudio realizado por (Mäntylä, 2017) una revisión de literatura académica solo puede centrarse en un área particular del análisis de sentimientos,

ya que generalmente incluye entre 10 y 100 estudios. Hasta 2017 se habían publicado casi 7000 artículos sobre este tema y que abarcaban las más distintas áreas de interés como el comercio, la política, la educación, la salud y en general todo aquello donde se pueda recopilar y analizar grandes volúmenes de información vertidas en la web, pudiéndose obtener patrones que permiten predecir el comportamiento de los usuarios de una forma bastante acertada.

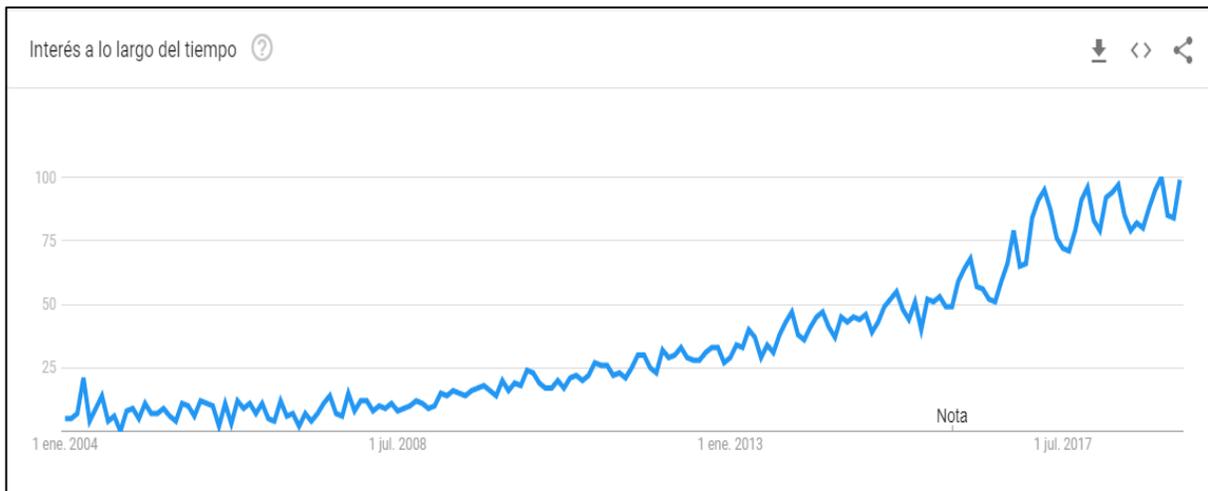


Ilustración 5: Muestra aumento de interés en búsquedas en Google sobre el tema.

4. DESARROLLO DEL MODELO

4.1. Recolección de Datos.

Considerando el objetivo principal del presente proyecto de título, se determinó establecer como dominio de la investigación una cantidad suficiente de comentarios realizados en la red social Twitter, relativos a opiniones en idioma español referidas a mujeres y en los que se pudiese apreciar algún tipo de animadversión que pudiera ser categorizada como agresivo o no agresivo en contra de las féminas. Considerándose como condición previa el hecho de que no

exista ningún tipo de sesgo al recopilar los comentarios, ya sea de género, estrato social, nivel cultural, grupo etario, etc., ya que la violencia de género no atiende ningún tipo de clasificación en aquellos que la ejercen. Estos datos son anónimos y fueron publicados entre los años 2015 y 2018.

Para construir el corpus utilizado fue necesario extraer comentarios de la red social mencionada, lo cual fue realizado usando en primera instancia una API que permitía a personas que habían aplicado como desarrolladores en Twitter y que tuvieran ciertos privilegios (Tokens), descargar determinadas cantidades de comentarios de acuerdo con patrones de palabras en idioma español (referidos a mujeres), a un archivo .txt que posteriormente debía ser llevado a formato Excel para su etiquetado. Este método fue luego descartado en atención a que los comentarios descargados resultaban en gran cantidad repetidos o se hallaban cortados o inconclusos alterando el sentido de su emisión. En segundo término, se continuó la extracción de comentarios, realizando búsquedas avanzadas de comentarios a través de una cuenta en la red social Twitter, ingresando patrones de búsqueda de palabras o frases que ciertamente se relacionaran con el tema, extrayendo manualmente los contenidos para luego vaciarlos al archivo en formato Excel complementando con los extraídos con el primer método. Mediante los procesos señalados anteriormente, se logró extraer un total de 1494 comentarios.

Para que el corpus pudiera ser entrenado y puesto a prueba por los clasificadores, era necesario etiquetar cada comentario con la polaridad correspondiente al análisis de sentimiento que se pretende desarrollar, con especial cuidado de que dicho proceso sea lo más objetivo posible. Para ello el corpus fue dividido en seis sets de 250 comentarios cada uno, los que fueron etiquetados por 3 evaluadores humanos distintos, en total participaron 18 voluntarios (7 mujeres y 11 hombres), en un rango etario de entre 17 y 81 años, quienes clasificaron los 1494 comentarios en dos categorías de polaridad (**Agresivo** y **No Agresivo**) respectivamente. La Tabla 1 muestra un extracto del corpus a modo de ejemplo.

Ingeniería de Ejecución en Computación e Informática - UBB

COMENTARIOS	EVALUADOR 1	EVALUADOR 2	EVALUADOR 3
“venga, vamos a brindar como cuando nos trajeron las pizzas a la academia; vamos a brindar por haber tenido la maravillosa oportunidad de haber podido vivir una maravillosa experiencia con personas maravillosas”	NO AGRESIVO	NO AGRESIVO	NO AGRESIVO
agarré el guante del novio tonta hueona A128	AGRESIVO	AGRESIVO	AGRESIVO
Con razón apareció está Rata de la Orsini defendiendo a Salamanca en el Mentiras Verdaderas	AGRESIVO	AGRESIVO	NO AGRESIVO
Consideraciones tácticas y estratégicas.. Dice Boric .. Habla como todo un estratega.. Pero no le funciona	NO AGRESIVO	NO AGRESIVO	NO AGRESIVO
Hasta cuando a este punga comunacho con corte de pelo de sicario se le permite hacer lo que quiera ?	NO AGRESIVO	AGRESIVO	NO AGRESIVO
Muy mal Boric y Orsini. Aún no toman el peso de estar en política representando una nueva forma de hacer las cosas.	NO AGRESIVO	NO AGRESIVO	NO AGRESIVO
Orsini cayó en la trampa d boric jaja una pendeja q no le ha ganado a nadie!	AGRESIVO	AGRESIVO	NO AGRESIVO

Tabla 1: Ejemplo de set de comentarios y etiquetado de evaluadores.

4.2. Limpieza de Datos.

Para realizar el proceso de limpieza del corpus, en primer lugar se agruparon en un solo archivo en formato Excel los distintos sets evaluados por los apreciadores, para luego detectar visualmente aquellos comentarios que estuvieran repetidos y que fueron eliminados manualmente, a continuación y utilizando un programa en lenguaje de programación Java, desarrollado por investigadores del Grupo SOMOS de la Universidad del Bio Bio y adaptado a la necesidades de nuestra investigación, se logró de manera automatizada realizar la lectura del corpus en Excel, ejecutando los procesos de limpieza de puntuación, gramaticales y ortográficos, como asimismo aplicar los features elegidos; esto de acuerdo a las herramientas disponibles mencionadas en el apartado 3.4. Toda vez que el proceso de clasificación automática y obtención de métricas puede ser realizado con la herramienta WEKA, siendo necesario que el programa genere finalmente un archivo ARFF entregando el corpus ya procesado.

4.3. Elección de Algoritmos.

Recordemos que se mencionó en el numeral 3.3. que el enfoque automático utiliza algoritmos de aprendizaje que pueden ser supervisados y no supervisados. Los más utilizados son los

supervisados, debido a la forma de trabajo, que permite verificar los resultados de clasificación y así poder estudiar el rendimiento del algoritmo, que es parte del objetivo de esta investigación.

Los algoritmos supervisados de clasificación automática pueden ser divididos en clasificadores Generativos o Discriminativos (también llamados condicionales). Los clasificadores generativos aprenden el modo a partir de la probabilidad conjunta $P(C, D)$ (Donde D es el documento y C la clase) y realizan la predicción maximizando la probabilidad condicional $P(C|D)$ que se calcula utilizando el teorema de Bayes. A diferencia de estos, los clasificadores discriminativos modelan directamente la probabilidad condicional $P(C|D)$ para realizar la predicción, es decir, a partir de la estructura oculta de los datos sin tener en cuenta la forma en que se generan. La desventaja de los algoritmos generativos es que resuelven un problema más general como paso intermedio, en cambio, los algoritmos discriminativos resuelven el problema de clasificación directamente.

Según la literatura estudiada, el algoritmo generativo más utilizado es el algoritmo Naive Bayes, y dentro de los modelos discriminativos más relevantes se encuentra el algoritmo de Árboles de Decisión, Random Forest y el algoritmo Support Vector Machine (SVM).

4.4. Creación del Corpus (Acuerdo Kappa).

El coeficiente **Kappa de Cohen** es una medida del acuerdo entre dos evaluadores que determinan a qué categoría pertenece un número finito de sujetos, por lo que se elimina un acuerdo debido al azar. Los dos calificadores o un acuerdo en su calificación (es decir, la categoría a la que se asigna un tema) o no están de acuerdo; no hay grados de desacuerdo (es decir, no hay ponderaciones).

El coeficiente kappa (κ) toma valores entre -1 y +1; mientras más cercano a +1, mayor es el grado de concordancia inter-observador. Por el contrario, un valor de $\kappa = 0$ refleja que la

concordancia observada es precisamente la que se espera a causa exclusivamente del azar. Valoración del coeficiente Kappa (Landis & Koch, 1977).

Desde	Hasta	Fuerza de la concordancia
0,00		Pobre (Poor)
0,01	0,20	Leve (Slight)
0,21	0,40	Aceptable (Fair)
0,41	0,60	Moderada (Moderate)
0,61	0,80	Considerable (Substantial)
0,81	1,00	Casi perfecta (Almost perfect)

Tabla 2: Rangos de acuerdo en coeficiente Kappa de Cohen.

Definición: Si p_a = la proporción de observaciones en concordancia y p_ϵ = la proporción en concordancia debido al azar, entonces **la kappa de Cohen** es:

$$\kappa = \frac{p_a - p_\epsilon}{1 - p_\epsilon}$$

O alternativamente,

$$\kappa = \frac{n_a - n_\epsilon}{n - n_\epsilon}$$

Ecuación 6: Definiciones matemáticas de coeficiente Kappa de Cohen.

donde n = número de sujetos, n_a = número de acuerdos y n_ϵ = número de acuerdos debido al azar.

El índice Kappa de Cohen fue aplicado para determinar el nivel de concordancia entre cada uno de los evaluadores, para lo cual se utilizó el complemento Real Statistics para la Planilla de Calculo Excel (Zaiontz, 2015), proceso que consistió en analizar mediante el uso de Tablas Dinámicas, un set de comentarios etiquetados por voluntarios evaluadores respecto de que si éstos eran **Agresivos** o **No Agresivos** contra la mujer según sus apreciaciones, obteniéndose

Ingeniería de Ejecución en Computación e Informática - UBB

los niveles de acuerdo entre los evaluadores, para luego aplicar la herramienta que arroja el índice Kappa, el error estándar, y los límites superior e inferior del rango con el que se obtuvo el coeficiente, finalmente como este mismo proceso se realizó en tres pares de evaluadores (De acuerdo al esquema diagramado en la ilustración 5 y la tabla 3).

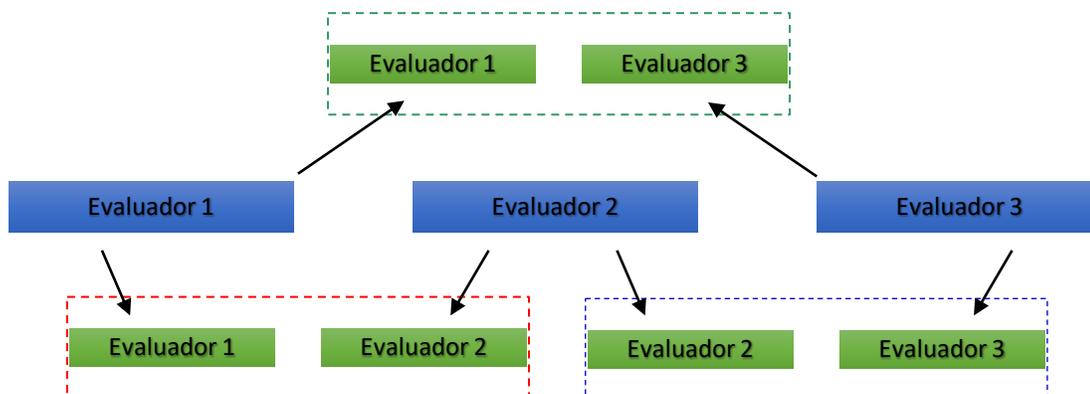


Ilustración 6: Diagrama con pares de evaluadores para coeficiente Kappa de Cohen.

COMENTARIOS	EVAL 1	EVAL 2	EVAL 3	CLASIFICADOR FINAL
“venga, vamos a brindar como cuando nos trajeron las pizzas a la academia; vamos a brindar por haber tenido la maravillosa oportunidad de haber podido vivir una maravillosa experiencia con personas maravillosas”	NO AGRESIVO	NO AGRESIVO	NO AGRESIVO	NO AGRESIVO
agarré el guante del novio tonta hueona A128	AGRESIVO	AGRESIVO	AGRESIVO	AGRESIVO
Con razón apareció está Rata de la Orsini defendiendo a Salamanca en el Mentiras Verdaderas	AGRESIVO	AGRESIVO	NO AGRESIVO	AGRESIVO
Consideraciones tácticas y estratégicas.. Dice Boric .. Habla como todo un estratega.. Pero no le funciona	NO AGRESIVO	NO AGRESIVO	AGRESIVO	NO AGRESIVO
Hasta cuando a este punga comunacho con corte de pelo de sicario se le permite hacer lo que quiera ?	NO AGRESIVO	AGRESIVO	NO AGRESIVO	NO AGRESIVO
Muy mal Boric y Orsini. Aún no toman el peso de estar en política representando una nueva forma de hacer las cosas.	NO AGRESIVO	NO AGRESIVO	NO AGRESIVO	NO AGRESIVO
Orsini cayó en la trampa d boris jaja una pendeja q no le ha ganado a nadie!	AGRESIVO	AGRESIVO	NO AGRESIVO	AGRESIVO

Tabla 3: Ejemplo de corpus con sus evaluadores y proceso de clasificación final.

Tras realizar el proceso señalado anteriormente, el corpus quedó compuesto por 1494 comentarios, y al aplicar la herramienta Real Statistics a los distintos pares de evaluadores, se

Ingeniería de Ejecución en Computación e Informática - UBB

promediaron los resultados obteniéndose un Índice Kappa Promedio equivalente a **0,84044296**, lo que da un nivel de fuerza de la concordancia **casi perfecta**, el detalle de las tablas de confusión y los resultados arrojados por la herramienta se hallan disponibles en el Anexo 1.

Ciertamente se puede señalar que el valor del coeficiente Kappa es considerado muy aceptable, estimándose que lo anterior se debe a que en este experimento se solicitó a los evaluadores que se pronunciaran sólo respecto de la polaridad de los comentarios, es decir que indicaran si éstos eran **Agresivos** o **No Agresivos** hacia las mujeres, pudiéndose concluir de la simple observación que existe un alto nivel de concordancia entre los evaluadores humanos de distintos grupos etarios, lo que simplemente refleja que el tema de la violencia de género es una problemática suficientemente conocida y definida por la sociedad en nuestro país y en el resto del mundo. La tabla 4 muestra para el ejemplo referido el resultado final de este proceso.

COMENTARIO	CLASIFICADOR
“venga, vamos a brindar como cuando nos trajeron las pizzas a la academia; vamos a brindar por haber tenido la maravillosa oportunidad de haber podido vivir una maravillosa experiencia con personas maravillosas”	NO AGRESIVO
agarré el guante del novio tonta hueona A128	AGRESIVO
Con razón apareció está Rata de la Orsini defendiendo a Salamanca en el Mentiras Verdaderas	AGRESIVO
Consideraciones tácticas y estratégicas.. Dice Boric .. Habla como todo un estratega.. Pero no le funciona	NO AGRESIVO
Hasta cuando a este punga comunacho con corte de pelo de sicario se le permite hacer lo que quiera?	NO AGRESIVO
Muy mal Boric y Orsini. Aún no toman el peso de estar en política representando una nueva forma de hacer las cosas.	NO AGRESIVO
Orsini cayó en la trampa d boris jaja una pendeja q no le ha ganado a nadie!	AGRESIVO

Tabla 4: Ejemplo de extracto de corpus resultante.

Finalmente, y luego de efectuar una revisión manual para descartar comentarios que se hallaban repetidos, que solo contenían caracteres con signos sin significado y emoticones, se puede señalar que las estadísticas del corpus (Sin considerar la etiqueta de clasificación de cada comentario), son las siguientes:

- Cantidad de comentarios: 1470 comentarios.
- Cantidad de palabras totales: 33.012 palabras.
- Cantidad de caracteres utilizados: 156.055 caracteres.

4.5. Descripción del Corpus.

Como bien se ha señalado en el numeral 4.1. del presente trabajo, el corpus corresponde a un set de comentarios recopilados desde twitter mediante dos métodos distintos latamente explicados en el mencionado apartado, datos que fueron vaciados en una planilla de cálculo Excel para proceder a su clasificación manual por parte de tres evaluadores, respecto de la polaridad de los comentarios en relación a si eran Agresivo o No Agresivo contra las mujeres.

Realizado el proceso anterior y agrupados todos los comentarios en un solo archivo Excel, se eliminaron los archivos repetidos y que no tuvieran un significado para la investigación, enseguida se determinó el coeficiente Kappa de Cohen para verificar el nivel de acuerdo entre los evaluadores, también ya explicado en el numeral que antecede, para luego de verificar que el nivel de fuerza de concordancia era suficiente, se procedió a dejar como clasificador final la polaridad con mayor recurrencia en la fila, siendo entonces la mayor la definitiva para cada comentario.

Una vez definido el corpus final en el archivo Excel, se estuvo en condiciones de proceder a la limpieza y creación de los distintos corpus que se utilizarían en los experimentos, de acuerdo al siguiente detalle:

- Corpus limpio sin Stop Words y sin POS
- Corpus limpio con Stop Words y sin POS
- Corpus limpio sin Stop Words y con POS
- Corpus limpio con Stop Words y con POS

Para ello se utilizó un programa en lenguaje Java (Elgueta, 2017), desarrollado por investigadores del grupo de investigación SoMoS (Software, Modelling & Science) de la

Ingeniería de Ejecución en Computación e Informática - UBB

Universidad del Bio Bio y adaptado a las necesidades de nuestra investigación, que a continuación será explicado a grandes rasgos en su funcionamiento:

Primero es necesario para poder utilizar correctamente el programa, la instalación de las siguientes librerías:

- Apache POI: Para realizar la lectura del archivo en formato Excel.
- Stanford NLP Tagger: Para aplicar el etiquetado POS para idioma español.

Para realizar la lectura del archivo que contiene al corpus original en formato Excel, el programa permite buscar el archivo en el equipo gracias a una interfaz gráfica implementada con un JFileChooser. A continuación, también usando una interfaz gráfica, el programa permite al usuario asignar una ruta de destino donde se guardará el archivo de salida en formato ARFF.

Siguiendo el proceso, el programa permite definir en el archivo de salida la cabecera del archivo considerando el formato necesario para que los archivos ARFF y de acuerdo a los atributos que se van a almacenar para que puedan ser analizados por la herramienta WEKA.

Luego, utilizando los métodos provistos por Apache POI, se realiza la lectura del Excel para la creación de los distintos corpus, recordando que éstos están conformados por 2 columnas, siendo la primera el comentario y la segunda la clasificación de la polaridad. Ubicándose luego en una fila, se lee la columna correspondiente al comentario, aplicándose los filtros de limpieza y la selección de features de acuerdo a la versión del corpus que se desea crear para la clasificación. Se continúa con la columna de clasificación, en la cual sólo se aplica el filtro de limpieza, agregándose al archivo de salida en el apartado de set de datos. Este proceso se itera para cada una de las filas disponibles en el corpus, por último, se cierra el archivo de salida y finaliza el programa.

Para que se ejecuten los filtros de limpieza mencionados anteriormente, se desarrollaron los métodos que se describen a continuación:

filterComillasEspacios: Permite eliminar todo tipo de comillas, además de eliminar el espacio doble o superior. El filtro es usado en la creación de todas las versiones de corpus.

```
private static String filterComillasEspacios(String stringValue){
    String output = stringValue.replaceAll("\\\"", "")
        .replaceAll("\\'", "").replaceAll(""", "").replaceAll("&quot;", "")
        .replaceAll("&quot;", "").replaceAll("&quot;", "").replaceAll("&quot;", "")
        .replaceAll("&quot;", "").replaceAll("&quot;", "");
    while(output.contains("  ")){
        output = output.replaceAll("  ", " ");
    }
    return output;
}
```

Ilustración 9: Método para eliminar comillas y espacios dobles.

filterTildes: Permite reemplazar letras con tildes, diéresis, ñ y caracteres especiales de letras por su homólogo sin caracteres especiales. El filtro es usado en la creación de todas las versiones de corpus.

```
private static String filterTildes(String stringValue) {
    String output = stringValue.replaceAll("á", "a")
        .replaceAll("é", "e")
        .replaceAll("í", "i")
        .replaceAll("ó", "o")
        .replaceAll("ú", "u")
        .replaceAll("Á", "A")
        .replaceAll("É", "E")
        .replaceAll("Í", "I")
        .replaceAll("Ó", "O")
        .replaceAll("Ú", "U")
        .replaceAll("ü", "u")
        .replaceAll("Û", "U")
        .replaceAll("ñ", "n")
        .replaceAll("Ñ", "N")
        .replaceAll("ã", "a")
        .replaceAll("Ã", "A");
    return output;
}
```

Ilustración 10: Método para reemplazar caracteres especiales.

filterSignos: Permite eliminar los signos de puntuación. El filtro es usado en la creación de todas las versiones de corpus.

```
private static String filterSignos(String stringCellValue) {
    return stringCellValue.replaceAll("\\.", "").replaceAll(":", "")
        .replaceAll("...", "").replaceAll(",", "");
}
```

Ilustración 11: Método para eliminar signos de puntuación.

filterExclInt: permite eliminar los signos de exclamación e interrogación. El filtro es usado en la creación de todas las versiones de corpus.

```
private static String filterExclInt(String stringCellValue) {
    return stringCellValue.replaceAll("¿", "").replaceAll("\\?", "")
        .replaceAll(";", "").replaceAll("!", "");
}
```

Ilustración 12: Método para eliminar signos de exclamación, interrogación y otros.

filterStopWords: Permite eliminar los StopWords leídos desde archivo, los cuales se encuentran posteriormente almacenados en un ArrayList. Este filtro es utilizado solamente para generar las versiones 2 y 4. En las versiones 1 y 3 no es utilizado.

```
private static String filterStopWords(String stringCellValue, ArrayList<String> stopWordsVector) {
    ArrayList<String> stopLower = new ArrayList<>();
    String output = " "+stringCellValue+" ";
    for(String stopword: stopWordsVector){
        stopLower.add(stopword.toLowerCase());
        output = output.replaceAll(" "+stopword+" ", " ");
    }
    for(String stopword2: stopLower){
        output = output.replaceAll(" "+stopword2+" ", " ");
    }
    return output.substring(1, output.length()-1);
}
```

Ilustración 13: Método para eliminar las Stop Words.

La idea de realizar la limpieza por medio de métodos independientes responde a la necesidad de poder aplicar estos filtros según la configuración de limpieza que se requiera, esto dado por

el contexto, los documentos y la necesidad de estudio del investigador pensando en trabajos futuros.

Por otro lado, para poder aplicar el feature correspondiente al etiquetado POS, es necesario configurar en el programa la herramienta provista por el Stanford POS Tagger, utilizando los modelos para el idioma español. Para ello es necesario realizar lo siguiente:

- ✓ En la dirección o ruta, donde se ubica el nuevo proyecto se crea una carpeta con el nombre de taggers (puede tener otro nombre).
- ✓ Descomprimir archivos provistos por Stanford ([//nlp.stanford.edu/software/tagger.shtml](http://nlp.stanford.edu/software/tagger.shtml)) y ubicarlos en la carpeta creada en el paso anterior, estos archivos tienen la extensión tagger y props.
- ✓ Importar la librería para dejar disponible el modelo en el programa.
- ✓ Concluida la configuración, codificar el método principal en el que constará un constructor de la clase MaxentTagger, al cual se le tiene que enviar como parámetro un archivo entrenado, que en este caso corresponde al tagger en español.
- ✓ Por último, se etiqueta la cadena a procesar.

Para poder entender de mejor manera el funcionamiento del Stanford POS Tagger, en la ilustración 13 se presenta un ejemplo en el cual se etiqueta una cadena de prueba:

```
import edu.stanford.nlp.tagger.maxent.MaxentTagger;

public class Tagger {
    public static void main(String[] args){
        //Iniciar el tagger
        MaxentTagger tagger = new MaxentTagger("taggers/spanish.tagger");
        // Ejemplo de una cadena
        String ejemplo = "Esta es una oración de prueba";
        // Cadena
        String taggeo = tagger.tagString(ejemplo);
        // Presentamos el resultado
        System.out.println(taggeo);
    }
}
```

Ilustración 14: Ejemplo de funcionamiento de Stanford POS Tagger.

Al momento de ejecutar este código en uno de los comentarios del corpus elaborado, el resultado es el que aparece en la tabla 5.

Corpus original	Qué bueno, entonces habrá que dar dos grandes PLR !!!	Agresivo
Corpus 3	Que_pe000000 bueno_aq0000 entonces_rg habrá_vaif000 que_cs	Agresivo
Sin SW con POS	dar_vmn0000 dos_dn0000 grandes_aq0000 PLR_np00000	

Tabla 5: Ejemplo de corpus antes y después de etiquetado POS.

4.6. Preprocesamiento.

Fue desarrollado metódicamente a partir de la implementación de la totalidad los numerales precedentes del presente capítulo, como ya se ha señalado la construcción de un clasificador de textos basado en un sistema de aprendizaje automático consta de varias etapas secuenciales. En primer lugar, es necesario preparar los datos del corpus para entrenar los algoritmos. Para ello, se debe limpiar y normalizar su información con el objetivo de reducir o eliminar aquellos datos que puedan influir de manera negativa en el resultado final. A continuación, cada uno de los textos de ejemplo se somete a un proceso denominado *tokenización*, el cual los divide en unidades más pequeñas o *tokens* y que habitualmente son las palabras de los mensajes. A partir de los *tokens* se extraen las características que representen a los mensajes originales y, de manera opcional, se puede aplicar un método para reducir su número, para así finalmente con ellas proceder a entrenar los clasificadores. Todo este proceso se llevó a cabo con el programa en lenguaje java señalado en el numeral anterior, entregando los distintos corpus en formato ARFF en condiciones de ser analizado por la herramienta WEKA. La tabla 6 muestra un ejemplo de comentario preprocesados para los distintos corpus.

CORPUS	COMENTARIO	CLASIFICADOR
Corpus original	Qué bueno, entonces habrá que dar dos grandes PLR !!!	Agresivo
Corpus 1 Sin SW sin POS	Que bueno entonces habra que dar dos grandes PLR	Agresivo
Corpus 2 Con SW sin POS	Que habrá PLR	Agresivo
Corpus 3 Sin SW con POS	Que_pe000000 bueno_aq0000 entonces_rg habrá_vaif000 que_cs dar_vmn0000 dos_dn0000 grandes_aq0000 PLR_np00000	Agresivo
Corpus 4 Sin SW sin POS	Que_pr000000 habrá_vaif000 PLR_np00000	Agresivo

Tabla 6: Ejemplo de las distintas versiones de corpus.

5. EXPERIMENTACION

5.1. Introducción.

La experimentación en esta investigación tiene por finalidad principal, el poder comparar los resultados arrojados por la herramienta de análisis WEKA, en el proceso de aplicación de los cuatro algoritmos de clasificación elegidos, es decir, Árboles de decisión, Naive Bayes, Máquinas de Soporte Vectorial y Random Forest, en un set de comentarios extraídos desde la red social Twitter, que fue etiquetado de forma especial por un grupo de evaluadores voluntarios, clasificándolo de acuerdo a la apreciación de polaridad de ser **Agresivos** o **No Agresivos** contra las mujeres. Igualmente, se pretende observar la incidencia que plantea la aplicación del etiquetado gramatical POS (Part of Speech) y la técnica de eliminación de Stop Words en el rendimiento de los clasificadores mencionados.

5.2. Descripción de los experimentos.

El **primer experimento** plantea la aplicación de un proceso de análisis de sentimiento sobre el corpus elaborado y descrito en el apartado 4.4. y 4.5. Este desarrollo considera una etapa de

limpieza del corpus que consiste en la aplicación de un programa en lenguaje Java, el cual en primera instancia lee los comentarios desde un archivo Excel, para luego aplicar filtros programados para eliminar puntuación, acentos, comillas, signos de exclamación o interrogación, además de varios caracteres especiales encontrados normalmente en los Twits. Continuando este proceso se efectúa la selección de features POS o eliminación de Stop Words según corresponda, para finalmente proporcionar un archivo con extensión ARFF, preprocesado y estructurado para ser analizado por el programa WEKA. Este procedimiento genera según sea el caso, cuatro corpus distintos con las características que a continuación se mencionan:

- **Corpus sin POS sin Stop words**
- **Corpus sin POS con Stop words**
- **Corpus con POS sin Stop words**
- **Corpus con POS con Stop words**

El análisis con la herramienta WEKA (University of Waikato, NZ, 2018) se inicia activando el programa eligiendo la modalidad de exploración que se presenta en la ventana, acto seguido en la pantalla preproceso siguiente se carga el archivo ARFF preparado, luego se elige el filtro en este caso usamos el StringToWordVector que convierte los atributos de tipo String en un conjunto de atributos representando la ocurrencia de las palabras del texto. En seguida pasamos a la pestaña Clasificar donde en la ventana clasificador elegimos nuevamente el filtro y seleccionamos el algoritmo de clasificación que utilizaremos, finalmente seleccionamos el proceso Cross-validation con el que Weka realizará una validación cruzada estratificada del número de particiones dado (Folds). Esta validación consiste en que dado un número n se divide los datos en n partes y, por cada parte, se construye el clasificador con las $n-1$ partes restantes y se prueba con esa. Así por cada una de las n particiones, en nuestro caso utilizaremos el valor de 10 Folds que viene por defecto. Como resultado del análisis con la herramienta WEKA obtendremos un resumen con información del uso del clasificador seleccionado y las métricas. Este mismo proceso se debe repetir con los cuatro corpus mencionados anteriormente (García, Diego).

El **segundo experimento** contempla igualmente un análisis de sentimiento como el primero, pero difiere en la limpieza del set de comentarios obtenidos de Twitter, en atención a que se modificó el programa en lenguaje Java para que no se apliquen los filtros programados para eliminar puntuación, acentos, comillas, signos de exclamación o interrogación, además de varios caracteres especiales encontrados normalmente en los Twits, aplicándose en consecuencia solamente el feature POS y la eliminación de StopWords, de este proceso se obtendrán también cuatro corpus como en el experimento anterior, lo que nos permitirá observar si este pequeño cambio generará alguna mejora en el rendimiento de los clasificadores. El análisis de los nuevos corpus con la herramienta WEKA se llevará a efecto en los mismos términos señalados en el primer experimento.

Cabe mencionar que los resultados obtenidos con la herramienta WEKA, serán procesados mediante tablas y gráficos y analizados posteriormente. El proceso general a realizar en la etapa de experimentación queda diagramado en la ilustración 14.

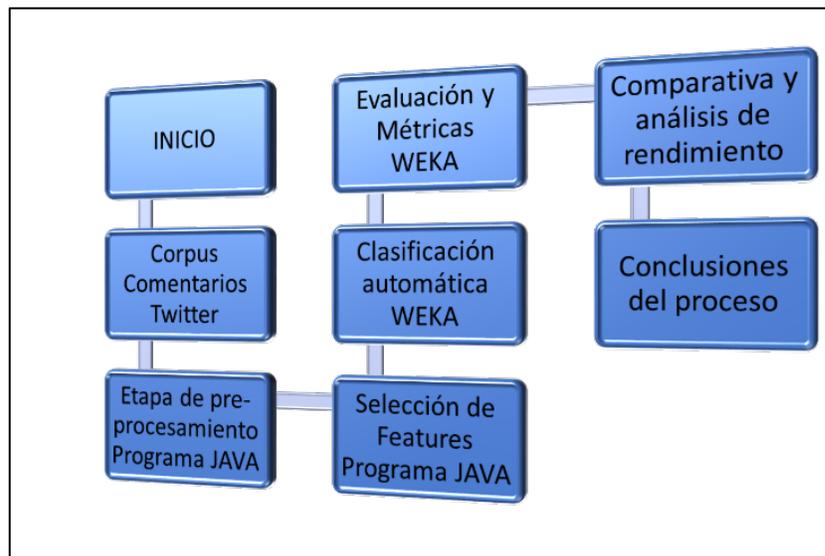


Ilustración 15: Diagrama con proceso etapa de experimentación

6. RESULTADOS Y ANALISIS DE LAS EXPERIMENTACIONES

6.1. Experimento N° 1.

- Corpus N° 1 sin POS sin Stop words.
- Corpus N° 2 sin POS con Stop words.
- Corpus N° 3 con POS sin Stop words.
- Corpus N° 4 con POS con Stop words.

RENDIMIENTO DE CLASIFICADORES								
	Decisión Tree		Naive Bayes		SVM		Random Forest	
	% Correcto Clasificados	% Incorrecto Clasificados						
CORPUS 1	83.54%	16.46%	86.46%	13.54%	84.83%	15.17%	88.91%	11.09%
	1228	242	1271	199	1247	223	1307	163
CORPUS 2	85.85%	14.15%	86.60%	13.40%	85.92%	14.08%	89.46%	10.54%
	1262	208	1273	197	1263	207	1315	155
CORPUS 3	82.24%	17.76%	84.42%	15.58%	82.79%	17.21%	88.03%	11.97%
	1209	261	1241	229	1217	253	1294	176
CORPUS 4	80.20%	19.80%	84.97%	15.03%	84.69%	15.31%	88.03%	11.97%
	1179	291	1249	221	1245	225	1294	176

Tabla 7: Rendimiento de clasificadores en Experimento N° 1.

Respecto del rendimiento de los clasificadores (Tabla 7), se aprecia que **RF** se comporta mejor en todos los corpus de prueba, con resultados correctos entre 88,03% y 89,46% el más alto (corpus 2), seguido por **NB**, **SVM** y **DT** respectivamente, este último cuenta con el más bajo en el corpus 4 equivalente a un 80,20%, al sacar un promedio entre el mayor y menor rendimiento se obtuvo una media de 84,83% correspondiente a **SVM** en el corpus 1.

Se puede observar que el rendimiento de los clasificadores aplicados sobre el corpus procesado con StopWords (corpus 2) aumentó levemente respecto al rendimiento obtenido cuando se aplicó al corpus original (corpus 1). Este aumento se observó en **DT** un 2% el más alto, en 1% incremento en **SVM** y **RF**, mientras que se mantuvo el porcentaje en **NB**.

Ingeniería de Ejecución en Computación e Informática - UBB

Respecto de la aplicación de etiquetado POS al Corpus 3, el clasificador basado en **NB** y **SVM** muestra una baja de rendimiento de -2% respecto del Corpus original (corpus 1), en tanto el clasificador basado en **DT** disminuyó en -1%, mientras que **RF** fue el de mejor rendimiento evidenciando una leve baja que no alcanza medio punto porcentual.

Respecto del rendimiento la aplicación de StopWords y etiquetas POS al corpus 4, comparado con los resultados del corpus original, se pudo apreciar que **RF** y **SVM** disminuyeron levemente menos de un punto porcentual, seguido de cerca por el clasificador **NB** con algo menos de -2%, en último lugar y con el porcentaje más bajo hallamos al **DT** con una disminución de -3%.

PARÁMETROS DE MÉTRICAS= Precision – Recall – F-Mesure												
	Decisión Tree			Naive Bayes			SVM			Random Forest		
%	Precisión	Recall	F-Mesure	Precisión	Recall	F-Mesure	Precisión	Recall	F-Mesure	Precisión	Recall	F-Mesure
CORPUS 1	0,835	0,835	0,834	0,865	0,865	0,864	0,848	0,848	0,848	0,895	0,889	0,887
CORPUS 2	0,868	0,859	0,855	0,868	0,866	0,864	0,859	0,859	0,859	0,899	0,895	0,893
CORPUS 3	0,822	0,822	0,821	0,844	0,844	0,844	0,828	0,828	0,828	0,887	0,880	0,878
CORPUS 4	0,813	0,802	0,795	0,851	0,850	0,848	0,846	0,847	0,846	0,885	0,880	0,878

Tabla 8: Resultados de métricas Precision, Recall y F-Mesure en Experimento N° 1.

Analizando los resultados obtenidos de las métricas Precision, Recall y F-Mesure (Tabla 8), para todos los corpus del experimento se observa que los mejores resultados son obtenidos utilizando el clasificador **RF**, que se comporta mejor en todos los corpus de prueba y obtiene las mejores métricas en el corpus 2 (0,899% - 0,895% - 0,893%), seguido por **NB**, **SVM** y **DT** respectivamente, este último cuenta con el más bajo en el corpus 4 equivalentes a (0,813% - 0,802% - 0,795%), al sacar un promedio entre las mejores y peores métricas se obtuvo una media aproximada cercana al 0,848% correspondiente a **SVM** en el corpus 1.

Por último, las matrices de confusión que se muestran a continuación (Tabla 9), nos permiten apreciar que sin lugar a dudas el clasificador **RF** alcanzó los mejores rendimientos al arrojar

que 482 comentarios **Agresivo** fueron clasificados correctamente como **Agresivo** (Verdaderos positivos) y 823 comentarios **No Agresivo** fueron clasificados correctamente como **No Agresivo** (Verdaderos negativos) eso es 1315 correctamente clasificados (corpus 2), mientras los peores resultados son de **DT** 383 y 796 respectivamente (corpus 4).

MATRICES DE CONFUSION DE CORPUS ANALIZADOS CON WEKA								
	Decision Tree		Naive Bayes		SVM		Random Forest	
	A	B	A	B	A	B	A	B
Corpus 1								
A= Agresivo	470	145	487	128	500	115	482	133
B= No Agresivo	97	758	71	784	108	747	30	825
Corpus 2								
A= Agresivo	442	173	477	138	503	112	492	123
B= No Agresivo	35	820	59	796	95	760	32	823
Corpus 3								
A= Agresivo	453	162	487	128	486	129	471	144
B= No Agresivo	99	756	101	754	124	731	32	823
Corpus 4								
A= Agresivo	383	232	469	146	490	125	477	138
B= No Agresivo	59	796	75	780	100	755	38	817

Tabla 9: Resumen de matrices de confusión en Experimento N° 1.

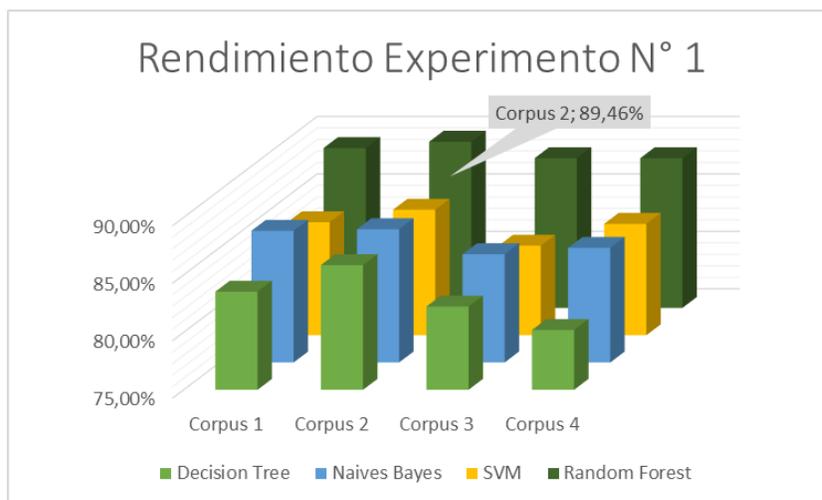


Ilustración 16: Gráfico con rendimiento de clasificadores en Experimento N° 1.

6.2. Experimento N° 2

- Corpus N° 1 sin POS sin Stop words.
- Corpus N° 2 sin POS con Stop words.
- Corpus N° 3 con POS sin Stop words.
- Corpus N° 4 con POS con Stop words.

RENDIMIENTO DE CLASIFICADORES								
	Decisión Tree		Naive Bayes		SVM		Random Forest	
	% Correcto Clasificados	% Incorrecto Clasificados						
CORPUS 1	83.27%	16.73%	86.05%	13.95%	85.03%	14.97%	88.30%	11.70%
	1224	246	1265	205	1250	220	1298	172
CORPUS 2	84.83%	15.17%	86.80%	13.20%	85.92%	14.08%	88.71%	11.29%
	1247	223	1276	194	1225	401	1304	166
CORPUS 3	82.45%	17.55%	84.08%	15.92%	83.40%	16.60%	87.69%	12.31%
	1212	258	1236	234	1226	244	1289	181
CORPUS 4	81.29%	18.71%	85.37%	14.63%	85.03%	14.97%	88.71%	11.29%
	1195	275	1255	215	1250	220	1304	166

Tabla 10: Rendimiento de clasificadores en Experimento N° 2.

Respecto del rendimiento de los clasificadores (Tabla 10), se aprecia que **RF** se comporta mejor en todos los corpus de prueba, con resultados correctos entre 87,69% y 88,71% el más alto (corpus 2 y 4), seguido por **NB**, **SVM** y **DT** respectivamente, este último cuenta con el más bajo en el corpus 4 equivalente a un 81,29%, al sacar un promedio entre el mayor y menor rendimiento se obtuvo una media de 85,03% correspondiente a **SVM** en el corpus 1.

Se puede observar que el rendimiento de los clasificadores aplicados sobre el corpus procesado con StopWords (corpus 2) aumentó levemente respecto al rendimiento obtenido cuando se aplicó al corpus original (corpus 1). Este aumento se observó en **DT** un 1,5% el más alto, en casi 1% incremento en **SVM** y **NB**, mientras subió medio punto porcentual en **RF**.

Ingeniería de Ejecución en Computación e Informática - UBB

Respecto de la aplicación de etiquetado POS al Corpus 3, el clasificador basado en **DT** y **RF** muestra una baja de rendimiento de casi -1% respecto del Corpus original (corpus 1), en tanto el clasificador basado en **SVM** disminuyó en -1,5%, mientras que **NB** fue el de peor rendimiento evidenciando una leve baja que no alcanza -2%.

Respecto del rendimiento la aplicación de StopWords y etiquetas POS al corpus 4, comparado con los resultados del corpus original, se pudo apreciar que **RF** y **NB** disminuyeron levemente alrededor de medio punto porcentual, en tanto que el clasificador **SVM** mantuvo exactamente el mismo porcentaje, mientras en último lugar y con el porcentaje más bajo aparece **DT** con una disminución de -2%.

PARÁMETROS DE MÉTRICAS= Precision – Recall – F-Mesure												
	Decisión Tree			Naive Bayes			SVM			Random Forest		
%	Precision	Recall	F-Mesure	Precision	Recall	F-Mesure	Precision	Recall	F-Mesure	Precision	Recall	F-Mesure
CORPUS 1	0,834	0,833	0,831	0,860	0,861	0,860	0,850	0,850	0,850	0,890	0,883	0,881
CORPUS 2	0,852	0,848	0,846	0,869	0,868	0,867	0,859	0,859	0,859	0,891	0,887	0,885
CORPUS 3	0,824	0,824	0,824	0,840	0,841	0,840	0,834	0,834	0,834	0,882	0,877	0,875
CORPUS 4	0,812	0,813	0,811	0,853	0,854	0,853	0,851	0,850	0,850	0,891	0,887	0,885

Tabla 11: Resultados de métricas Precision, Recall y F-Mesure en Experimento N° 2.

Analizando los resultados obtenidos de las métricas Precisión, Recall y F-Mesure (Tabla 11), para todos los corpus del experimento se observa que los mejores resultados son obtenidos utilizando el clasificador **RF**, que se comporta mejor en todos los corpus de prueba y obtiene las mejores métricas en el corpus 4 (0,891% - 0,887% - 0,885%), seguido por **NB**, **SVM** y **DT** respectivamente, este último cuenta con el más bajo en el corpus 4 equivalentes a (0,812% - 0,813% - 0,811%), al sacar un promedio entre las mejores y peores métricas se obtuvo una media aproximada cercana al 0,850% correspondiente a **SVM** en el corpus 1.

Por último, las matrices de confusión que se muestran a continuación (Tabla 12), nos permiten apreciar que sin lugar a dudas el clasificador **RF** alcanzó los mejores rendimientos al arrojar

Ingeniería de Ejecución en Computación e Informática - UBB

que 485 comentarios **Agresivo** fueron clasificados correctamente como **Agresivo** (Verdaderos positivos) y 819 comentarios **No Agresivo** fueron clasificados correctamente como **No Agresivo** (Verdaderos negativos) eso es 1304 correctamente clasificados (corpus 2 y 4), mientras los peores resultados son de **DT** 448 y 747 respectivamente (corpus 4).

MATRICES DE CONFUSION DE CORPUS ANALIZADOS CON WEKA								
	Decision Tree		Naive Bayes		SVM		Random Forest	
	A	B	A	B	A	B	A	B
Corpus 1								
A= Agresivo	452	163	491	124	504	111	471	144
B= No Agresivo	83	772	81	774	109	746	28	827
Corpus 2								
A= Agresivo	454	161	487	128	512	103	485	130
B= No Agresivo	62	793	66	789	104	751	36	819
Corpus 3								
A= Agresivo	466	149	488	127	492	123	472	143
B= No Agresivo	109	746	107	748	121	734	38	817
Corpus 4								
A= Agresivo	448	167	487	128	508	107	485	130
B= No Agresivo	108	747	87	768	113	742	36	819

Tabla 12: Resumen de matrices de confusión en Experimento N° 2.

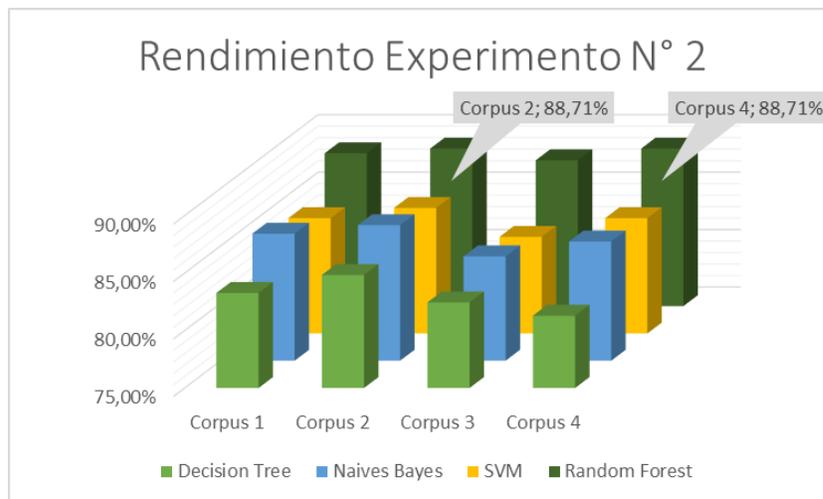


Ilustración 17: Gráfico con rendimiento de clasificadores en Experimento N° 2.

7. APLICACIÓN DESARROLLADA PARA USO DEL MODELO.

El prototipo de aplicación que se propone fue desarrollado en lenguaje Python y se trata de un programa que realiza el análisis de un corpus que ha sido previamente preprocesado y que a continuación será explicado a grandes rasgos en su funcionamiento:

Primero es necesario para poder utilizar el programa, se requiere de la instalación de las siguientes librerías:

- WX Python: Permite el desarrollo rápido de aplicaciones gráficas multiplataforma.
- NumPy: Es una librería para realizar cálculo numérico en Python, principalmente para crear y modificar matrices, además de hacer operaciones sobre ellas.
- Pandas: es una librería de Python destinada al análisis de datos, que proporciona unas estructuras de datos flexibles y que permiten trabajar con ellos de forma muy eficiente.
- SkLearn: es una biblioteca para aprendizaje de máquina de software libre para el lenguaje de programación Python. Incluye varios algoritmos de clasificación, regresión y análisis de grupos entre los cuales están máquinas de vectores de soporte, bosques aleatorios, Gradient boosting,

A continuación, mediante la función preparar se carga el archivo que contiene el corpus en formato .csv con los datos de la encuesta que será analizada y que ya fue preprocesado. Esto debe ser modificado en el main cuando se utilice un nuevo corpus.

```
def preparar(self):
    self.addToLog("Preparando datos...")
    data = pd.read_csv(
        'conSWconPOS.csv',
        # sep=';;;',
        engine='python'
    )
```

Ilustración 18: Función que carga el corpus al programa.

Ingeniería de Ejecución en Computación e Informática - UBB

Siguiendo con la ejecución, el programa realiza una limpieza donde se eliminan las comillas, se cambian las palabras escritas en mayúscula por minúsculas y se aplica comillas simples a los comentarios.

Luego encontramos el método y la función que define los vectores de entrenamiento y prueba que se realizarán a los datos contenidos en el corpus, en nuestro caso corresponde a un 70% del corpus para entrenamiento y el 30% para las pruebas, tomándose los datos de manera aleatoria.

```
xtrain, xtest, ytrain, ytest = tts(corpus, data['B'], test_size=0.30)
```

Ilustración 19: Método que define % de entrenamiento y pruebas.

```
def train_test_vector(self, xtrain, xtest):
    global vectorizer
    vectorizer = TfidfVectorizer(
        tokenizer=self.tokenize, max_df=0.80, use_idf=True, min_df=1)
    # vectorizer = CountVectorizer(min_df=1, binary=True) # métrica binaria
    vector_train = vectorizer.fit_transform(xtrain)
    vector_test = vectorizer.transform(xtest)
    return vector_train, vector_test, vectorizer
```

Ilustración 20: Función que realiza la vectorización de los datos.

Se procede luego con la aplicación de la función que ejecuta la tokenización del corpus analizado, que consiste en dividir los comentarios en unidades más pequeñas que en general se condicen con las mismas palabras que contienen.

```
def tokenize(self, text):
    tokens = nltk.word_tokenize(text, language='spanish')
    #tokens = nltk.word_tokenize(text, language='english')

    stems = []
    for item in tokens:
        stems.append(nltk.PorterStemmer().stem(item))
    return stems
```

Ilustración 21: Función que Tokeniza los textos ingresados.

Ingeniería de Ejecución en Computación e Informática - UBB

Luego encontramos la función donde se aplica el modelo elegido, en este caso hemos utilizado el Support Vector Machine incluido en el módulo sklearn SVM de la biblioteca NumPy para Python.

```
global modelo
modelo = svm.SVC(kernel='linear')
modelo.fit(X=xtrain, y=ytrain)
```

Ilustración 22: Método aplica modelo SVM del módulo sklearn de biblioteca Pandas.

Enseguida encontramos el método donde se aplica el modelo de predictivo y la entrega de un reporte con las métricas obtenidas con los resultados del proceso de clasificación realizado en el corpus.

```
prediccion = modelo.predict(xtest)

self.addToLog(pd.crosstab(ytest, prediccion, rownames=[
| 'REAL'], colnames=['PREDICCION']).to_string())
self.addToLog("")
self.addToLog(classification_report(ytest, prediccion))
```

Ilustración 23: Método donde se aplica el modelo predictivo y su respectivo reporte.

Finalmente, las métricas con los resultados obtenidos del análisis del corpus son mostradas en una interfaz gráfica que además contempla el ingreso de nuevos comentarios, los que son evaluados con una función que permite su clasificación a través del mismo modelo de predicción utilizado, imprimiendo en pantalla tanto el texto analizado, como la predicción resultante del proceso.

```
def procesar(self, texto):
    Xnew = [texto]
    vector_new = vectorizer.transform(Xnew)
    ynew = modelo.predict(vector_new)
    for i in range(len(Xnew)):
        self.addToLog("X=%s, Predicted=%s" % (Xnew[i], ynew[i]))
```

Ilustración 24: Función que analiza nuevos comentarios y los clasifica.

Ingeniería de Ejecución en Computación e Informática - UBB

```
def __init__(self, parent, title):
    print("Creando interfaz grafica...")
    wx.Frame.__init__(self, parent=parent, title=title, size=(900, 500))
```

Ilustración 25: Función para crear interfaz gráfica.

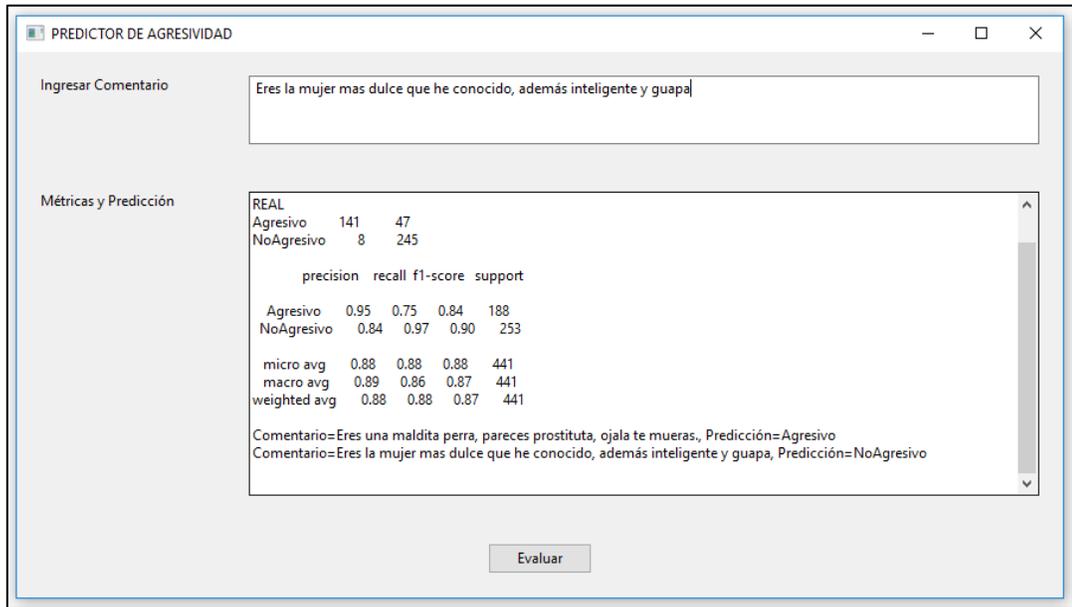


Ilustración 26: Ejemplo de interfaz gráfica generada por el programa.

8. CONCLUSIONES

8.1. Aportes.

Se puede afirmar que la principal contribución de esta investigación, es comprobar que este tipo de tecnología no solo sirve para un uso comercial, sino que puede ser utilizado en soluciones que puedan ayudar a prevenir la ocurrencia de delitos de connotación social, como es el caso de la violencia de género, pudiendo incluso ser adaptado a otras áreas como el maltrato escolar, laboral, de adultos mayores, etc., además se logró determinar en la acotada experimentación con algunos de los clasificadores ya existentes, cuáles de estos métodos son los que mejor se comportan, en cuanto a su rendimiento y certeza en la predicción de comentarios en un corpus en español, lo que ciertamente puede sentar un precedente a considerar en futuras investigaciones sobre el particular, especialmente teniendo en cuenta que en nuestro país existe muy poco desarrollo de este tema..

Conteste con lo anterior, a través del presente trabajo se entrega una especie de breve radiografía que nos muestra un análisis actualizado del estado del arte sobre esta materia. También desde un enfoque práctico, se investigó, experimentó y comentó el uso de una serie de herramientas necesarias en el desarrollo de este tipo de soluciones, adaptando los procedimientos existentes a nuestras necesidades, elaborándose además un programa básico con una interfaz simple que permite acceso a información de análisis y predicción.

El desarrollo de este proyecto conllevó la elaboración de un corpus que contiene los comentarios en español recolectados desde la red social Twitter, junto con su correspondiente etiqueta de polaridad asignada por los evaluadores humanos voluntarios que clasificaron cada texto. Lo anterior es absolutamente necesario para poder desarrollar las etapas de entrenamiento y prueba de los procesos de experimentación que plantea el aprendizaje automático. Finalmente, esta investigación implica también un punto de partida para futuros trabajos o

proyectos, que mediante la utilización de las técnicas de minería de opinión puedan entregar soluciones a otros problemas sociales en nuestro país.

8.2. Contraste de lo realizado con los objetivos planteados.

Este trabajo permitió comparar técnicas de aprendizaje automático aplicados al análisis de sentimiento sobre comentarios en español extraídos de la red social Twitter. A continuación, se presentan los objetivos específicos y se detallan sus cumplimientos:

OE1: Estudiar la literatura y trabajos relacionados con la detección automática de agresividad hacia la mujer, que utilicen técnicas de sentic computing. El trabajo realizado presenta una adecuada investigación desde el estudio del marco teórico de conceptos utilizados en análisis de sentimiento, hasta la búsqueda y consideración de herramientas necesarias para aplicar estas técnicas.

OE2: Recolectar opiniones publicadas en Twitter y construir un corpus, que contenga opiniones etiquetadas y que sirva para el entrenamiento y aprendizaje del modelo. Efectivamente se realizaron las gestiones tendientes a la recopilación de un set de comentarios desde la mencionada red social, los que fueron evaluados y etiquetados por personas voluntarias de distintos grupos etarios, todo ello ciertamente referido al tema de violencia de género que nos convocaba, ya que orientados a otros temas no prestan ninguna utilidad.

OE3: Aplicar técnicas de machine learning para la construcción de un clasificador de opiniones agresivas contra la mujer, analizando características utilizadas en la clasificación y las métricas de rendimiento obtenidas. Todo ello fue realizado en dos experimentos que contemplaban la clasificación y el análisis de los corpus creados con distintas características, para de esta forma obtener las métricas resultantes que nos permitirían evaluar y concluir al respecto.

OE4: Construir un software que sirva de prototipo y que permita validar los resultados obtenidos del modelo. Finalmente, también se elaboró un prototipo de programa que pudiera desarrollar el análisis de sentimiento a un corpus, entregando luego resultados con las métricas y por último permita realizar de manera automática la clasificación de un nuevo comentario.

8.3. Conclusiones.

En cuanto a la experimentación realizada en el presente trabajo de investigación, se puede concluir que el mejor rendimiento y las métricas más favorables, fueron obtenidas aplicando el método de clasificación Random Forest en el corpus donde fueron eliminadas las Stop Words (Corpus 2), estimándose que el excelente rendimiento obtenido 89%, evidenciado también en los otros métodos de clasificación (todos sobre 80%), obedece al registro de un casi perfecto nivel de acuerdo entre los evaluadores humanos (0,84%, de índice Kappa), donde se apreció una elevada coincidencia en la clasificación de los comentarios, que se estima proviene de la claridad que existe en la opinión pública, al momento de clasificar una conducta agresiva o no agresiva hacia las mujeres. También se cree que el hecho de desarrollar un análisis de sentimiento en base a la polaridad de un atributo donde aparecen solo dos opciones para clasificar (Agresivo y No Agresivo), sin que existan más categorías de emociones como en un análisis de afecto, o se requiera de un nivel de granulado mayor para obtener un resultado más preciso que haga alguna otra diferencia, ciertamente influyeron también en la obtención de tan altos rendimientos.

En este mismo orden de cosas, pero en lo que respecta a la limpieza y aplicación de features en los corpus analizados, la experimentación arrojó leves variaciones porcentuales entre aquellos procesos que incluían el uso de etiquetado POS, la eliminación de Stop Words y también si existía incidencia en el uso o no uso de puntuación, comillas, caracteres o signos especiales, mayúsculas, etc., alcanzándose como máximo dos o tres puntos porcentuales de variación, estimando que ello no implicaría resultados más certeros en las predicciones, sin embargo de

todas formas se recomienda usar cualquier proceso que ayude a mejorar los rendimientos de los clasificadores.

Por otra parte, en lo que respecta al programa prototipo de análisis y predicción, se decidió finalmente que la mejor forma de elegir el método de clasificación a utilizar, era obteniendo una media en el rendimiento de los clasificadores, es decir el promedio entre el menor y el mayor rendimiento que en el caso en estudio nos dio 85% aproximadamente, y que corresponde al valor presentado por el clasificador Support Vector Machine (SVM) en ambos experimentos, determinándose en consecuencia que éste era el clasificador más recomendable para el programa prototipo de análisis y predicción de comentarios, destacándose el hecho de que en la pruebas realizadas en todos los corpus y con todos los algoritmos propuestos, es con el mencionado clasificador que se han registrado resultados con un rendimiento cercano al 90% sin que se hayan detectado errores en la predicción.

8.4. Trabajos Futuros.

Entonces, podemos presumir que en el futuro las áreas de aplicación del análisis de sentimientos seguirán creciendo y las técnicas de análisis de sentimientos tenderán a ser una parte estandarizada y adaptable a muchos servicios y productos. Se piensa que los métodos de investigación indudablemente mejorarán con los avances en el procesamiento del lenguaje natural y el aprendizaje automático. Asimismo, se cree que los métodos de análisis de sentimientos actualmente basados principalmente en texto migrarán también hacia otros métodos informáticos como la voz, el reconocimiento de imagen o incluso el análisis de neuromarcadores. Finalmente se estima que en nuestro país deberían aumentar los desarrollos en este tipo de materias, para de esa forma seguir la tendencia registrada a nivel global sobre el Análisis de Sentimiento.

Ingeniería de Ejecución en Computación e Informática - UBB

En lo referente a la detección automática de agresividad hacia la mujer en comentarios emitidos en redes sociales, ciertamente es un proyecto que podría desarrollarse en conjunto con las autoridades de gobierno encargadas de su prevención, estimándose que un trabajo conjunto con profesionales del área y las víctimas de este flagelo, permitirían la construcción de un corpus mucho más completo y eficiente. Además, se podría considerar realizar un estudio para estandarizar las herramientas de preprocesamiento que entreguen el mejor rendimiento para cada uno de los clasificadores.

Por último se estima que es muy recomendable poder gestionar con alguna organización relacionada con el tema, la obtención de respaldo y financiamiento para desarrollar un proyecto que permita la implementación efectiva de una aplicación de escritorio o móvil que contemple el uso de esta tecnología para detectar agresividad en comentarios de redes sociales o de algún otro medio (email, sms, etc.), funcionalidad que podría complementarse con información sobre violencia de género, sugerencias para víctimas y familiares, o incluso comunicación online para dirigir a los usuarios hacia oficinas o centros de atención. Todo lo anterior ciertamente se traduce en un beneficio social que, a través de un curso de acción real, propone un esfuerzo orientado a disminuir y tratar de erradicar este tipo de violencia.

9. REFERENCIAS.

- Biagioni, R. (2016). *The SenticNet sentiment lexicon: Exploring semantic richness in multi-word concepts* (Vol. 5). Spring.
- Bravo-Marquez, F., Mendoza, M., & Poblete, B. (Octubre de 2014). Meta-level sentiment models for big social data analysis. *Knowledge-Based Systems*, vol. 69, no 1, pp. 86–99. Obtenido de Knowledge-Based Systems: <https://doi.org/10.1016/j.knosys.2014.05.016>
- Breiman, L. (1999). *Random forests*, UC Berkeley. Obtenido de http://machinelearning202.pbworks.com/w/file/attach/60606349/breiman_randomforests.pdf
- Cambria, E., & Hussain, A. (2012). *Sentic computing: Techniques, tools, and applications* (Vol. 2). Springer Science & Business Media.
- Caton, S., Hall, M., & Weinhardt, C. (Diciembre de 2015). How do politicians use Facebook? An applied Social Observatory. *Big Data & Society*, vol. 2, no 2. Obtenido de <https://doi.org/10.1177%2F2053951715612822>
- Congreso Nacional , Chile. (2005). *Ley N° 20.066 de Violencia Intrafamiliar*. (C. Nacional, Ed.) Obtenido de <https://www.leychile.cl/Navegar?idNorma=242648>
- Córdova , A. (2015). *Diseño y construcción de un sistema web de análisis de opiniones en Twitter integrando algoritmos de data mining*. Santiago: Universidad de Chile, Tesis de Pregrado.
- Da Silva, N., Hruschka, E., & Hruschka, E. (Octubre de 2014). Tweet sentiment analysis with classifier ensembles. *Decision Support Systems*, Volume 66, Pages 170-179. Obtenido de <https://doi.org/10.1016/j.dss.2014.07.003>
- Dang-Xuan, L., Stieglitz, S., Wladarsch, J., & Neuberger, C. (Junio de 2013). An Investigation of Influentials and the Role of Sentiment in Political Communication on Twitter during Election Periods. *Information Communication and Society*, 5, 795–825. Obtenido de https://www.researchgate.net/publication/259704131_An_Investigation_of_Influentials_and_the_Role_of_Sentiment_in_Political_Communication_on_Twitter_during_Election_Periods
- Elgueta, J. (2017). *Comparación de rendimiento de técnicas de aprendizaje automático para análisis de afecto sobre textos en español*. Concepción: Universidad del Bio Bio, Tesis de Magister.
- García, Diego. (s.f.). *Manual de WEKA*. Obtenido de <https://knowledgesociety.usal.es/sites/default/files/MANUAL%20WEKA.pdf>
- Guo, L., & Vargo, C. (Julio de 2015). The Power of Message Networks: A Big-Data Analysis of the Network Agenda Setting Model and Issue Ownership. *Mass Communication and Society*, 18(5), 557-576. Obtenido de <https://doi.org/10.1080/15205436.2015.1045300>
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. (2009). *The WEKA data mining software: an update* (Vol. 11). ACM SIGKDD Explorations Newsletter.

- Hastie, T., Friedman, J., & Tibshirani, R. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.
- Hernández, N. (2016). *Metodología para el diseño y construcción de un lexicón de opinión, basado en comentarios de Twitter aplicado al proyecto "OpinionZoom"*. Santiago: Universidad de Chile, Tesis de Pregrado.
- INDH, Chile. (2009). *Instituto Nacional de Derechos Humanos*. Obtenido de <https://www.indh.cl/quienes-somos/>
- Jungherr, A. (2015). *Analyzing Political Communication with Digital Trace Data: The Role of Twitter Messages in Social Science Research*. Springer International Publishing Switzerland.
- Landis, R., & Koch, G. (1977). An Application of Hierarchical Kappa-type Statistics in the Assessment of Majority Agreement among Multiple Observers. *International Biometric Society*, 33(2), 363-374.
- Liu, B. (2012). *Sentiment analysis and opinion mining, Synthesis lectures on human language technologies*. Morgan & Claypool Publishers. Obtenido de <https://doi.org/10.2200/S00416ED1V01Y201204HLT016>
- Mäntylä, M. (2017). *The evolution of sentiment analysis—A review of research topics, venues, and top cited papers*. Finlandia: Institute of Software Technology, University of Stuttgart.
- Martínez Camara, E. (2015). *RUJA, Repositorio Universidad de Jaén*. (U. d. Jaén, Ed.) Obtenido de <http://hdl.handle.net/10953/727>
- Medhat, W., Hassan, A., & Korashy, H. (Diciembre de 2014). *Sentiment analysis algorithms and applications: A survey*. Ain Shams Engineering Journal, Vol. 5, N° 4, pp. 1093–1113. Obtenido de <https://doi.org/10.1016/j.asej.2014.04.011>
- Mejova, Y. (2009). *Sentiment analysis: An overview; University of Iowa, Department, Computer Science*. Obtenido de https://www.researchgate.net/profile/Yelena_Mejova/publication/264840229_Sentiment_Analysis_An_Overview/links/590ad68e0f7e9b1d0823eff2/Sentiment-Analysis-An-Overview.pdf
- Montecinos, L. (2014). *Análisis de sentimientos y predicción de eventos en twitter*. Santiago: Universidad de Chile, Tesis de Pregrado.
- Naciones Unidas. (1993). Declaración sobre la eliminación de la violencia contra la mujer. *Asamblea General, 85a. Sesión Plenaria*. Nueva York. Obtenido de <https://www.ohchr.org/sp/professionalinterest/pages/violenceagainstwomen.aspx>
- Organización Mundial de la Salud. (2013). *Organización Mundial de la Salud*. (O. P. Salud, Ed.) Obtenido de http://www.who.int/entity/reproductivehealth/topics/violence/vaw_series/es/index.html
- Oyarzún, C. (2014). *Análisis automático de sentimientos sobre opiniones y/o comentarios de novelas en español*. Concepción: Universidad del Bio Bio, Tesis de Pregrado.
- Ponce de León, F. (2015). *Uso de la ingeniería de negocios en diseño e implementación de negocio para Start up basada en Web Opinion Mining*. Santiago: Universidad de Chile, Tesis de Postgrado.

- Prata, D., Soares, K., Silva, M., Trevisan, D., & Letouze, P. (2016). “Social Data Analysis of Brazilian’s Mood from Twitter”. *International Journal of Social Science and*, 6(3), 179-183. Obtenido de <http://www.ijssh.org/vol6/640-H013.pdf>
- Quinlan, J. (1993). *Programs for Machine Learning*. Morgan Kaufmann Publishers.
- Scherer, K. R. (1984). *Emotion as a multicomponent process: A model and some cross-cultural data* (Vol. 5). *Review of Personality & Social Psychology*. Obtenido de <https://psycnet.apa.org/record/1986-17269-001>
- Sobrino, J. (2018). *Análisis de sentimientos en Twitter, TFM*. Universitat Oberta de Catalunya. Obtenido de <http://hdl.handle.net/10609/81435>
- University of Waikato, NZ. (2018). *Downloading and installing Weka*. Obtenido de <https://www.cs.waikato.ac.nz/ml/weka/downloading.html>
- Vapnik, V. (1982). *Theory of Probability & Its Applications* (Vol. 26). Society for Industrial and Applied Mathematics.
- Violencia contra la mujer en Chile y derechos humanos. (2017). *Informe Temático*. Obtenido de uchile.cl: http://www.uchile.cl/documentos/descarga-el-informe-violencia-contra-la-mujer-en-chile-y-derechos-humanos-pdf-29-mb_141701_0_2237.pdf
- Yu, Y., & Wang, X. (Julio de 2015). World Cup 2014 in the Twitter World: A big data analysis of sentiments in U.S. sports fans’ tweets. *Computers in Human Behavior*, Vol. 48, pp. 392–400. Obtenido de <https://doi.org/10.1016/j.chb.2015.01.075>
- Zaiontz, C. (2015). *Real Statistics Using Excel*. Obtenido de <http://www.real-statistics.com/>

10. ANEXOS

Anexo 1: Matrices de confusión para cálculo de Coeficiente Kappa de Cohen.

NIVEL DE ACUERDO (EVALUADOR 1 – EVALUADOR 2)			
Etiquetas	AGRESIVO	NO AGRESIVO	Total general
AGRESIVO	593	26	619
NO AGRESIVO	112	762	874
Total general	705	788	1493

Tabla 13: Matriz de confusión (Evaluador 1 y Evaluador 2) índice Kappa.

Cohen's Kappa	
Alpha	0,05
kappa	0,813365195
std err	0,015072879
lower	0,783822896
upper	0,842907494

Tabla 14: Coeficiente Kappa (Evaluador 1 y Evaluador 2).

NIVEL DE ACUERDO (EVALUADOR 2 – EVALUADOR 3)			
Etiquetas	AGRESIVO	NO AGRESIVO	Total general
AGRESIVO	586	120	706
NO AGRESIVO	17	771	788
Total general	603	891	1494

Tabla 15: Matriz de confusión (Evaluador 2 y Evaluador 3) índice Kappa.

Cohen's Kappa	
Alpha	0,05
kappa	0,813365195
std err	0,015072879
lower	0,783822896
upper	0,842907494

Tabla 16: Coeficiente Kappa (Evaluador 2 y Evaluador 3).

Ingeniería de Ejecución en Computación e Informática - UBB

NIVEL ACUERDO (EVALUADOR 1 – EVALUADOR 3)			
Etiquetas	AGRESIVO	NO AGRESIVO	Total general
AGRESIVO	572	47	619
NO AGRESIVO	30	844	874
Total general	602	891	1493

Tabla 17: Matriz de confusión (Evaluador 1 y Evaluador 3) índice Kappa

Cohen's Kappa	
Alpha	0,05
kappa	0,89332518
std err	0,011834285
lower	0,870130408
upper	0,916519952

Tabla 18: Coeficiente Kappa (Evaluador 1 y Evaluador 3).

Tabla 19: Promedio coeficiente Kappa de Cohen.

INDICE DE KAPPA PROMEDIO
0,84044296

El coeficiente kappa (κ) toma valores entre -1 y +1; mientras más cercano a +1, mayor es el grado de concordancia inter-observador. Por el contrario, un valor de $\kappa = 0$ refleja que la concordancia observada es precisamente la que se espera a causa exclusivamente del azar. Valoración del coeficiente Kappa (Landis y Koch, 1977)	Desde	Hasta	Fuerza de la concordancia
	0,00		Pobre (Poor)
	0,01	0,20	Leve (Slight)
	0,21	0,40	Aceptable (Fair)
	0,41	0,60	Moderada (Moderate)
	0,61	0,80	Considerable (Substantial)
	0,81	1,00	Casi perfecta (Almost perfect)

Tabla 20: Rangos de concordancia en coeficiente Kappa de Cohen.

Anexo 2: Listado de Stops Words usados en la limpieza de los Corpus.

él	buena	dice	este	informó	ninguno	posible
ésta	buenas	dicen	esto	junto	ningunos	próximo
éstas	bueno	dicho	estos	la	no	próximos
éste	buenos	dieron	estoy	lado	nos	primer
éstos	cómo	diferente	estuvo	las	nosotras	primera
última	cada	diferentes	ex	le	nosotros	primero
últimas	casi	dijeron	existe	les	nuestra	primeros
último	cerca	dijo	existen	llegó	nuestras	principalmente
últimos	cierto	dio	explicó	lleva	nuestro	propia
a	cinco	donde	expresó	llevar	nuestros	propias
añadió	comentó	dos	fin	lo	nueva	propio
aún	como	durante	fue	los	nuevas	propios
actualmente	con	e	fuera	luego	nuevo	pudo
adelante	conocer	ejemplo	fueron	lugar	nuevos	pueda
además	consideró	el	gran	más	nunca	puede
afirmó	considera	ella	grandes	manera	o	pueden
agregó	contra	ellas	ha	manifestó	ocho	pues
ahí	cosas	ello	había	mayor	otra	qué
ahora	creo	ellos	habían	me	otras	que
al	cual	embargo	haber	mediante	otro	quedó
algún	cuales	en	habrá	mejor	otros	queremos
algo	cualquier	encuentra	hace	mencionó	para	quién
alguna	cuando	entonces	hacen	menos	parece	quien
algunas	cuanto	entre	hacer	mi	parte	quienes
alguno	cuatro	era	hacerlo	mientras	partir	quiere

Tabla 21: Set con Stop Words utilizados en limpieza de los corpus.

Ingeniería de Ejecución en Computación e Informática - UBB

algunos	cuenta	eran	hacia	misma	pasada	realizó
alrededor	da	es	haciendo	mismas	pasado	realizado
ambos	dado	esa	han	mismo	pero	realizar
ante	dan	esas	hasta	mismos	pesar	respecto
anterior	dar	ese	hay	momento	poca	sí
antes	de	eso	haya	mucha	pocas	sólo
apenas	debe	esos	he	muchas	poco	se
aproximadamente	deben	está	hecho	mucho	pocos	señaló
aquí	debido	están	hemos	muchos	podemos	sea
así	decir	esta	hicieron	muy	podrá	sean
aseguró	dejó	estaba	hizo	nada	podrán	según
aunque	del	estaban	hoy	nadie	podría	segunda
ayer	demás	estamos	hubo	ni	podrían	segundo
bajo	dentro	estar	igual	ningún	poner	seis
bien	desde	estará	incluso	ninguna	por	ser
buen	después	estas	indicó	ningunas	porque	será
serán	sin	su	tendrán	toda	tres	va
sería	sino	sus	tenemos	todas	tuvo	vamos
si	sobre	tal	tener	todavía	u	van
sido	sola	también	tenga	todo	un	varias
siempre	solamente	tampoco	tengo	todos	una	varios
siendo	solas	tan	tenido	total	unas	veces
siete	solo	tanto	tercera	tras	uno	ver
sigue	solos	tenía	tiene	trata	unos	vez
siguiente	son	tendrá	tienen	través	usted	y
ya	yo					

Tabla 22: Continuación set de Stop Words utilizados en limpieza de los corpus.

Anexo 3: Resultados de la clasificación entregados por WEKA.

• Corpus 1 – Clasificador Decision Tree.

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      1228           83.5374 %
Incorrectly Classified Instances    242           16.4626 %
Kappa statistic                    0.658
Mean absolute error                 0.2005
Root mean squared error             0.3707
Relative absolute error             41.1946 %
Root relative squared error         75.1443 %
Total Number of Instances          1470

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0,764   0,113   0,829     0,764   0,795     0,660   0,862    0,788    Agresivo
                0,887   0,236   0,839     0,887   0,862     0,660   0,862    0,863    NoAgresivo
Weighted Avg.   0,835   0,185   0,835     0,835   0,834     0,660   0,862    0,832

=== Confusion Matrix ===

  a  b  <-- classified as
470 145 | a = Agresivo
 97 758 | b = NoAgresivo
    
```

Ilustración 27: Anexo Resultados Clasificación WEKA - Corpus 1 Clasificador DT

• Corpus 1 – Clasificador Naive Bayes

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      1271           86.4626 %
Incorrectly Classified Instances    199           13.5374 %
Kappa statistic                    0.7182
Mean absolute error                 0.1494
Root mean squared error             0.3374
Relative absolute error             30.6925 %
Root relative squared error         68.3909 %
Total Number of Instances          1470

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0,792   0,083   0,873     0,792   0,830     0,721   0,909    0,882    Agresivo
                0,917   0,208   0,860     0,917   0,887     0,721   0,909    0,917    NoAgresivo
Weighted Avg.   0,865   0,156   0,865     0,865   0,864     0,721   0,909    0,902

=== Confusion Matrix ===

  a  b  <-- classified as
487 128 | a = Agresivo
 71 784 | b = NoAgresivo
    
```

Ilustración 28: Anexo Resultados Clasificación WEKA - Corpus 1 Clasificador NB

• **Corpus 1 – Clasificador Support Vector Machine.**

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      1247           84.8299 %
Incorrectly Classified Instances    223           15.1701 %
Kappa statistic                    0.6878
Mean absolute error                0.1517
Root mean squared error            0.3895
Relative absolute error            31.1696 %
Root relative squared error        78.9565 %
Total Number of Instances         1470

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0,813   0,126   0,822     0,813   0,818     0,688   0,843    0,747    Agresivo
                0,874   0,187   0,867     0,874   0,870     0,688   0,843    0,831    NoAgresivo
Weighted Avg.   0,848   0,162   0,848     0,848   0,848     0,688   0,843    0,796

=== Confusion Matrix ===

  a  b  <-- classified as
500 115 |  a = Agresivo
108 747 |  b = NoAgresivo
    
```

Ilustración 29: Anexo Resultados Clasificación WEKA - Corpus 1 Clasificador SVM

• **Corpus 1 – Clasificador Random Forest.**

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      1307           88.9116 %
Incorrectly Classified Instances    163           11.0884 %
Kappa statistic                    0.7667
Mean absolute error                0.2804
Root mean squared error            0.3313
Relative absolute error            57.609 %
Root relative squared error        67.1529 %
Total Number of Instances         1470

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0,784   0,035   0,941     0,784   0,855     0,775   0,927    0,922    Agresivo
                0,965   0,216   0,861     0,965   0,910     0,775   0,927    0,926    NoAgresivo
Weighted Avg.   0,889   0,140   0,895     0,889   0,887     0,775   0,927    0,924

=== Confusion Matrix ===

  a  b  <-- classified as
482 133 |  a = Agresivo
 30 825 |  b = NoAgresivo
    
```

Ilustración 30: Anexo Resultados Clasificación WEKA - Corpus 1 Clasificador RF

Ingeniería de Ejecución en Computación e Informática - UBB

• **Corpus 2 – Clasificador Decision Tree.**

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      1262           85.8503 %
Incorrectly Classified Instances    208            14.1497 %
Kappa statistic                    0.6998
Mean absolute error                0.2178
Root mean squared error            0.3435
Relative absolute error            44.7462 %
Root relative squared error        69.638 %
Total Number of Instances         1470

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0,719   0,041   0,927      0,719   0,810     0,714   0,861    0,849    Agresivo
                0,959   0,281   0,826      0,959   0,887     0,714   0,861    0,844    NoAgresivo
Weighted Avg.   0,859   0,181   0,868      0,859   0,855     0,714   0,861    0,846

=== Confusion Matrix ===
  a  b  <-- classified as
442 173 |  a = Agresivo
 35 820 |  b = NoAgresivo
    
```

Ilustración 31: Anexo Resultados Clasificación WEKA - Corpus 2 Clasificador DT

• **Corpus 2 – Clasificador Naive Bayes.**

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      1273           86.5986 %
Incorrectly Classified Instances    197            13.4014 %
Kappa statistic                    0.7196
Mean absolute error                0.151
Root mean squared error            0.3318
Relative absolute error            31.0244 %
Root relative squared error        67.2646 %
Total Number of Instances         1470

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0,776   0,069   0,890      0,776   0,829     0,724   0,908    0,886    Agresivo
                0,931   0,224   0,852      0,931   0,890     0,724   0,908    0,916    NoAgresivo
Weighted Avg.   0,866   0,159   0,868      0,866   0,864     0,724   0,908    0,904

=== Confusion Matrix ===
  a  b  <-- classified as
477 138 |  a = Agresivo
 59 796 |  b = NoAgresivo
    
```

Ilustración 32: Anexo Resultados Clasificación WEKA - Corpus 2 Clasificador NB

Ingeniería de Ejecución en Computación e Informática - UBB

• **Corpus 2 – Clasificador Support Vector Machine.**

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      1263           85.9184 %
Incorrectly Classified Instances    207           14.0816 %
Kappa statistic                    0.7095
Mean absolute error                0.1408
Root mean squared error            0.3753
Relative absolute error            28.9332 %
Root relative squared error        76.0713 %
Total Number of Instances         1470

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                0,818   0,111   0,841     0,818   0,829     0,710   0,853    0,764    Agresivo
                0,889   0,182   0,872     0,889   0,880     0,710   0,853    0,839    NoAgresivo
Weighted Avg.   0,859   0,152   0,859     0,859   0,859     0,710   0,853    0,808

=== Confusion Matrix ===

  a  b  <-- classified as
503 112 |  a = Agresivo
 95 760 |  b = NoAgresivo
    
```

Ilustración 33: Anexo Resultados Clasificación WEKA - Corpus 2 Clasificador SVM

• **Corpus 2 – Clasificador Random Forest.**

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      1228           83.5374 %
Incorrectly Classified Instances    242           16.4626 %
Kappa statistic                    0.658
Mean absolute error                0.2005
Root mean squared error            0.3707
Relative absolute error            41.1946 %
Root relative squared error        75.1443 %
Total Number of Instances         1470

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                0,764   0,113   0,829     0,764   0,795     0,660   0,862    0,788    Agresivo
                0,887   0,236   0,839     0,887   0,862     0,660   0,862    0,863    NoAgresivo
Weighted Avg.   0,835   0,185   0,835     0,835   0,834     0,660   0,862    0,832

=== Confusion Matrix ===

  a  b  <-- classified as
470 145 |  a = Agresivo
 97 758 |  b = NoAgresivo
    
```

Ilustración 34: Anexo Resultados Clasificación WEKA - Corpus 2 Clasificador RF

• **Corpus 3 – Clasificador Decision Tree**

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      1209           82.2449 %
Incorrectly Classified Instances    261           17.7551 %
Kappa statistic                    0.6299
Mean absolute error                 0.2165
Root mean squared error             0.3773
Relative absolute error             44.4849 %
Root relative squared error         76.4853 %
Total Number of Instances          1470

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0,737   0,116   0,821     0,737   0,776     0,632   0,857    0,799    Agresivo
                0,884   0,263   0,824     0,884   0,853     0,632   0,857    0,851    NoAgresivo
Weighted Avg.   0,822   0,202   0,822     0,822   0,821     0,632   0,857    0,829

=== Confusion Matrix ===

  a  b  <-- classified as
453 162 |  a = Agresivo
 99 756 |  b = NoAgresivo
    
```

Ilustración 35: Anexo Resultados Clasificación WEKA - Corpus 3 Clasificador DT

• **Corpus 3 – Clasificador Naive Bayes.**

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      1241           84.4218 %
Incorrectly Classified Instances    229           15.5782 %
Kappa statistic                    0.6779
Mean absolute error                 0.1753
Root mean squared error             0.3566
Relative absolute error             36.017 %
Root relative squared error         72.2855 %
Total Number of Instances          1470

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0,792   0,118   0,828     0,792   0,810     0,678   0,899    0,867    Agresivo
                0,882   0,208   0,855     0,882   0,868     0,678   0,899    0,912    NoAgresivo
Weighted Avg.   0,844   0,170   0,844     0,844   0,844     0,678   0,899    0,893

=== Confusion Matrix ===

  a  b  <-- classified as
487 128 |  a = Agresivo
101 754 |  b = NoAgresivo
    
```

Ilustración 36: Anexo Resultados Clasificación WEKA - Corpus 3 Clasificador NB

• **Corpus 3 – Clasificador Support Vector Machine.**

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      1217           82.7891 %
Incorrectly Classified Instances    253            17.2109 %
Kappa statistic                    0.646
Mean absolute error                 0.1721
Root mean squared error             0.4149
Relative absolute error             35.3628 %
Root relative squared error         84.1 %
Total Number of Instances          1470

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0,790   0,145   0,797     0,790   0,793     0,646   0,823    0,717    Agresivo
                0,855   0,210   0,850     0,855   0,852     0,646   0,823    0,811    NoAgresivo
Weighted Avg.   0,828   0,183   0,828     0,828   0,828     0,646   0,823    0,772

=== Confusion Matrix ===

  a  b  <-- classified as
486 129 |  a = Agresivo
124 731 |  b = NoAgresivo
    
```

Ilustración 37: Anexo Resultados Clasificación WEKA - Corpus 3 Clasificador SVM

• **Corpus 3 – Clasificador Random Forest.**

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      1294           88.0272 %
Incorrectly Classified Instances    176            11.9728 %
Kappa statistic                    0.7475
Mean absolute error                 0.2988
Root mean squared error             0.3483
Relative absolute error             61.3852 %
Root relative squared error         70.6129 %
Total Number of Instances          1470

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0,766   0,037   0,936     0,766   0,843     0,757   0,916    0,896    Agresivo
                0,963   0,234   0,851     0,963   0,903     0,757   0,916    0,920    NoAgresivo
Weighted Avg.   0,880   0,152   0,887     0,880   0,878     0,757   0,916    0,910

=== Confusion Matrix ===

  a  b  <-- classified as
471 144 |  a = Agresivo
 32 823 |  b = NoAgresivo
    
```

Ilustración 38: Anexo Resultados Clasificación WEKA - Corpus 3 Clasificador RF

Ingeniería de Ejecución en Computación e Informática - UBB

• **Corpus 4 – Clasificador Decision Tree.**

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      1179           80.2041 %
Incorrectly Classified Instances    291           19.7959 %
Kappa statistic                    0.5765
Mean absolute error                 0.2534
Root mean squared error            0.3706
Relative absolute error             52.0678 %
Root relative squared error        75.1297 %
Total Number of Instances         1470

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0,623   0,069   0,867     0,623   0,725     0,596   0,871    0,837    Agresivo
                0,931   0,377   0,774     0,931   0,845     0,596   0,871    0,885    NoAgresivo
Weighted Avg.   0,802   0,248   0,813     0,802   0,795     0,596   0,871    0,865

=== Confusion Matrix ===
  a  b  <-- classified as
383 232 |  a = Agresivo
 59 796 |  b = NoAgresivo
    
```

Ilustración 39: Anexo Resultados Clasificación WEKA - Corpus 4 Clasificador DT

• **Corpus 4 – Clasificador Naive Bayes.**

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      1294           88.0272 %
Incorrectly Classified Instances    176           11.9728 %
Kappa statistic                    0.7475
Mean absolute error                 0.2988
Root mean squared error            0.3483
Relative absolute error             61.3852 %
Root relative squared error        70.6129 %
Total Number of Instances         1470

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0,766   0,037   0,936     0,766   0,843     0,757   0,916    0,896    Agresivo
                0,963   0,234   0,851     0,963   0,903     0,757   0,916    0,920    NoAgresivo
Weighted Avg.   0,880   0,152   0,887     0,880   0,878     0,757   0,916    0,910

=== Confusion Matrix ===
  a  b  <-- classified as
471 144 |  a = Agresivo
 32 823 |  b = NoAgresivo
    
```

Ilustración 40: Anexo Resultados Clasificación WEKA - Corpus 4 Clasificador NB

• **Corpus 4 – Clasificador Support Vector Machine.**

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      1245           84.6939 %
Incorrectly Classified Instances    225           15.3061 %
Kappa statistic                     0.6837
Mean absolute error                 0.1531
Root mean squared error            0.3912
Relative absolute error             31.4491 %
Root relative squared error        79.3098 %
Total Number of Instances         1470

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0,797   0,117   0,831     0,797   0,813     0,684   0,840    0,747    Agresivo
                0,883   0,203   0,858     0,883   0,870     0,684   0,840    0,826    NoAgresivo
Weighted Avg.   0,847   0,167   0,846     0,847   0,846     0,684   0,840    0,793

=== Confusion Matrix ===
  a  b  <-- classified as
490 125 |  a = Agresivo
100 755 |  b = NoAgresivo
    
```

Ilustración 41: Anexo Resultados Clasificación WEKA - Corpus 4 Clasificador SVM

• **Corpus 4 – Clasificador Random Forest.**

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      1294           88.0272 %
Incorrectly Classified Instances    176           11.9728 %
Kappa statistic                     0.7482
Mean absolute error                 0.2295
Root mean squared error            0.3175
Relative absolute error             47.148 %
Root relative squared error        64.3722 %
Total Number of Instances         1470

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0,776   0,044   0,926     0,776   0,844     0,756   0,919    0,910    Agresivo
                0,956   0,224   0,855     0,956   0,903     0,756   0,919    0,916    NoAgresivo
Weighted Avg.   0,880   0,149   0,885     0,880   0,878     0,756   0,919    0,913

=== Confusion Matrix ===
  a  b  <-- classified as
477 138 |  a = Agresivo
 38 817 |  b = NoAgresivo
    
```

Ilustración 42: Anexo Resultados Clasificación WEKA - Corpus 4 Clasificador RF