Diseño de un aplicativo para el análisis de sentimiento de reseñas de películas

Juan Pablo Giraldo Mosquera[1]
[1]Pontificia Universidad Javeriana, Cl. 18, Cali – Valle del Cauca, Colombia
e-mail: juanpgym@gmail.com

Abstract – Product and Service reviews have a substantial value for companies in terms of the sentiment they convey about user experience. In the film industry, movie reviews are one of the most important topics for their customers in order to decide which movie is worth it to watch, that's why it is decide to carry out a project to handle the computational problem of automate the classification of sentiment in movie reviews, utilizing natural language processing and machine learning techniques to build a model capable of discerning sentiments expressed in reviews.

Keywords – Natural Language Processing, Machine Learning, Sentiment Analysis, Movie Reviews.

I. INTRODUCCIÓN

Internet, entre muchos de sus beneficios, otorga la facilidad de compartir experiencias y opiniones de todos sus usuarios. Basta con poseer acceso a una red social o cualquier plataforma en línea que permita publicar textos accesibles por toda la comunidad, en donde la producción textual de un individuo común tiene un alcance global de manera instantánea. Esto sumado al hecho de que este tipo de plataformas son accesibles por, prácticamente, cualquier persona, las convierte en la fuente más grande de información existente. Dicha información se ha convertido en un recurso invaluable para las organizaciones debido al fácil acceso que se tiene a ella y al impacto que genera en sus actividades comerciales, pues por medio de estas plataformas los consumidores están extendiendo su percepción sobre sus productos y servicios afectando (positiva o negativamente) a la imagen que se tiene de ellas en el mercado. Así mismo, el valor de esta información reside en la capacidad de poder identificar el sentimiento que se expresa en los textos publicados por los usuarios, sin embargo, dicha tarea se ha vuelto prácticamente imposible de realizar, por medios convencionales, debido al continuo crecimiento del volumen de los datos. Debido a esto, las organizaciones se han visto en la necesidad de automatizar esta tarea mediante técnicas que permitan la identificación de información relevante y el análisis de sentimiento de la misma para otorgarle un valor plausible que puedan aprovechar.

En este proyecto se construirán modelos de clasificación que permitan identificar el sentimiento de textos extraídos de reseñas de películas tomadas de la web IMDB (una de las más populares para este tipo consultas). Adicionalmente, se evaluará el desempeño de estos con el fin de determinar cuál de ellos posee un mejor desempeño basándose en métricas tangibles como la precisión y exactitud de los resultados. Así mismo, se busca implementar una aplicación que haga uso del modelo más prometedor y que permita a los usuarios aprovecharlo para poder decidir qué películas ver con un criterio más robusto al momento de invertir su dinero en plataformas de streaming o en salas de cine.

II. FUNDAMENTACIÓN TEÓRICA

La globalización de la comunicación a través de internet ha permitido que cualquier individuo sea capaz de plasmar su opinión sobre cualquier tópico a modo de texto en plataformas en línea como foros, redes sociales o portales de reseñas de clientes de distintas empresas. Es aquí entonces, donde el análisis de sentimiento entra en juego en algo conocido como minería de opinión que, si bién se pueden entender como sinónimos, éste último enfatiza no solamente en "darle un tono" o determinar el sentimiento de estas opiniones sino en cómo otorgar valor tangible a la tarea de clasificar una opinión como positiva o negativa. No es difícil deducir que el crecimiento exponencial de las opiniones que se comunican a través de internet ha causado que esta tarea de clasificación sea prácticamente imposible de realizar por el humano, por lo que desde la computación se ofrece una solución haciendo uso de las técnicas del aprendizaje automático [1]. Para el desarrollo de la fase experimental de este proyecto, se evaluó el desempeño de cuatro técnicas clásicas de aprendizaje y una técnica de aprendizaje profundo con el fin de seleccionar la técnica que brinda el mejor desempeño para la tarea de clasificación del sentimiento de reseñas de películas. A continuación, se exponen las técnicas consideradas en este proceso de selección:

K-Nearest Neighbors (K-NN)
Consiste en el almacenamiento y agrupación de los datos de entrada de la fase de entrenamiento según su clasificación (valor del atributo de interés) con el fin de usarlo en predicciones futuras. De manera ilustrativa, el proceso de predicción de este algoritmo consiste en clasificar un nuevo dato de entrada según la proximidad que este tenga con los datos previamente almacenados. [4] El valor de la clasificación finalmente, será la clasificación a la que pertenezcan la mayor cantidad de datos previos que estén más cercanos al entrante.

Logistic Regression
En un problema de clasificación binaria consta de identificar la probabilidad (p) de que un evento ocurra. Este algoritmo hace uso de una función "logit" que relaciona la probabilidad (número entre 0 y 1) a los datos que ingresan al modelo asignando la clasificación en función de si esta probabilidad está por encima (se clasifica en la clase positiva) o por debajo (se clasifica en a clase negativa) del umbral de selección (0.5 en este caso) [5]

Random Forest
Para comprender este algoritmo primero debe conocerse la definición de las unidades que lo componen también llamadas árboles de decisión. Un árbol de decisión es un algoritmo compuesto por nodos y ramas, donde los primeros se dividen en nodos raíz que representan decisiones sobre las variables relacionadas con el atributo objetivo (que se desea predecir) las cuales derivaran en dos o más eventos mutuamente excluyentes, seguidamente, los nodos internos representan una de las posibles opciones/decisiones en un punto determinado del árbol, y finalmente los nodos hoja representan el resultado final al que convergen las decisiones y eventos del árbol. [6]

Un "Random Forest" (o bosque aleatorio) es un clasificador que consiste en una colección de clasificadores estructurados como árboles de decisión en donde cada árbol depende de los valores de un vector aleatorio construido independientemente y con la misma distribución para todos los árboles en el bosque. Como se expuso anteriormente, cada árbol del bosque convergerá en un "voto" de clase en donde el veredicto de clasificación final se determinará por la clase con la mayor cantidad de votos, siendo esta la predicción del algoritmo. [7]

Support Vector Machines (SVM).
Es un algoritmo de aprendizaje automático supervisado que se puede utilizar para problemas de clasificación o regresión. Dadas 2 o más clases de datos etiquetadas, actúa como un clasificador discriminativo, definido formalmente por un hiperplano óptimo que separa todas las clases. Los nuevos ejemplos que luego se mapean en ese mismo espacio se pueden clasificar según el lado de la brecha en que se encuentran. La geometría nos dice que un hiperplano es un subespacio de una dimensión menos que su espacio ambiental. Por ejemplo, un hiperplano de un espacio ndimensional es un subconjunto plano con dimensión n - 1. Por su naturaleza, separa el espacio en dos medios espacios. Luego para una clasificación binaria el hiperplano correspondería a una recta que separa las clases. [9]

Recurrent Neural Network Long-Short Term Memory (LSTM)
Esta red surge como la solución al problema existente con la arquitectura básica de las redes recurrentes (RNN) que radica en la dispersión del gradiente, la cual genera una inestabilidad numérica que dificulta o imposibilita al algoritmo encontrar una solución óptima para el problema de clasificación debido a que el cálculo del gradiente nos permite encontrar la combinación de valores de los parámetros que minimizan la pérdida de la red, siendo esta la medida de cuan bien se ajusta el modelo a los datos de entrenamiento. Para ello, se opta por un tipo específico de red recurrente llamada LSTM (Long Short Term Memory) que facilita el proceso de memorización de la información pasada al entrenar el modelo usando backpropagation. En una red LSTM se añaden componentes conocidos como compuertas, estas añaden el factor de selectividad sobre cual información se mantiene en la red, descartando la información que no significa un aporte relevante al contexto de la secuencia de entrada. [11]

III. RESULTADOS

Tras la fase experimental en donde se busca la configuración de parámetros de los modelos que ofrece el mejor desempeño, se procede a realizar la evaluación de los modelos con la totalidad del corpus, midiendo su desempeño mediante los siguientes indicadores.

$$True\ Postive: TP \mid True\ Negative: TN \mid False\ Positive: FP \mid False\ Negative: FN$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}\ (1), \qquad Precision = \frac{TP}{TP + FP}\ (2), \qquad Recall = \frac{TP}{TP + FN}\ (3),$$

$$\text{F1Score} = \frac{Precision * Recall}{Precision + Recall}\ (4)$$

Los resultados de desempeño obtenidos para cada modelo se observan en la **Tabla 1**:

Tabla 1: Evaluación de Desempeño de los Modelos.

HYPERPARAMETER SEARCH & MODEL EVAL.					
Logistic Regression	**Accuracy:** 0.9010 **Precision:** 0.8946 **Recall:** 0.9089 **F1 Score:** 0.9017	***Random Forest***	**Accuracy:** 0.9460 **Precision:** 0.9414 **Recall:** 0.9513 **F1 Score:** 0.9463	**Bidirectional LSTM 128 Units**	**Accuracy:** 0.9311 **Precision:** 0.9228 **Recall:** 0.9509 **F1 Score:** 0.9169
Knearest Neighbors	**Accuracy:** 0.7981 **Precision:** 0.7830 **Recall:** 0.8244 **F1 Score:** 0.8032	***Support Vector Machine***	**Accuracy:** 0.9207 **Precision:** 0.9353 **Recall:** 0.9367 **F1 Score:** 0.9209		

Al observar los resultados obtenidos notamos que los modelos más prometedores corresponden al Random Forest y a la Red Bidireccional LSTM los cuales presentan valores cercanos en los indicadores de desempeño. Por tanto, se tiene en cuenta, como criterio adicional, el costo computacional (específicamente el tiempo de ejecución de su entrenamiento) para determinar si se tiene una decisiva en este aspecto.

Tabla 2: Tiempo de entrenamiento de los modelos con mejor desempeño.

Model	Mean Training Execution Time (10 Exec.)	Input Size
RNN (Bidirectional LSTM)	25.454 hrs	90000
Random Forest	45.427 min	90000

Como se observa en la tabla, el modelo de aprendizaje automático tiene un costo computacional de tiempo de ejecución de su entrenamiento mucho menor que el modelo de aprendizaje profundo. Teniendo en cuenta que el modelo se integrará a una aplicación que permita aprovecharlo, en el caso de posibles actualizaciones que requieran de un reentrenamiento, el Random Forest tiene la ventaja de poder ser reentrenado en solo el 0,03% del tiempo que tomaría el entrenamiento de la red Bidireccional LSTM.

IV. CONCLUSIONES

El proceso de experimentación para la selección del modelo más favorable con base en su desempeño y costo computacional nos permiten llegar a las siguientes conclusiones:

• Al lograr preparar el conjunto de datos de entrada concatenando y preprocesando las reseñas extraídas de IMDB, seleccionar las técnicas a explorar y comparar el desempeño las mismas, posteriormente seleccionar el modelo de la técnica más favorable que en este caso fue Random Forest, y finalmente implementar el aplicativo que permite a los usuarios aprovechar este modelo, finalmente se llega a que logran cumplirse los objetivos propuestos al inicio del proyecto.

• Realizar como primer paso la evaluación del desempeño de los modelos con sus parámetros por defecto aportó un punto de partida para la evaluación de desempeño, en especial para definir los rangos de búsqueda que se realizaron en la optimización de los parámetros de los modelos.

• Respecto a la búsqueda de hiper parámetros, para el caso del modelo Random Forest en donde dos de los parámetros resultantes de la búsqueda correspondían a los valores en los extremos de los rangos designados, notamos que al hacer una segunda búsqueda extendiendo el rango en dirección a esos extremos, encontramos otra configuración que mejoró el desempeño del modelo.

• Teniendo en cuenta que el modelo se integrará a una aplicación que permita aprovecharlo, en el caso de posibles actualizaciones que requieran de un reentrenamiento, el Random Forest tiene la ventaja de poder ser reentrenado en solo el 0,03% del tiempo que tomaría el entrenamiento de la red Bidireccional LSTM. Este criterio da una justificación más robusta sobre cual modelo realmente ofrece una ventaja adicional entre Random Forest y el método de aprendizaje profundo.

• En caso de querer usar el modelo en entornos más desafiantes, al menos, en términos de la longitud de las reseñas. Se requerirá de hacer un ajuste en la configuración inicial del vectorizador TF-IDF, el cual tiene una limitante del máximo de características a considerar de 10.000, en caso de que el modelo se use en un entorno en donde la longitud media exceda esta cantidad de palabras habrá parte de los documentos que no serán contempladas por el vectorizado.

V. BIBLIOGRAFÍA

[1] Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. En *Sentiment Analysis: A Fascinating Problem* (1.a ed., Vol. 1). Morgan & Claypool Publishers.

[2] Introduction to Machine Learning with Python (2016) Andreas C. Mueller and Sarah Guido

[3] Practical Statistics for Data Scientists. (2020) Peter Bruce, Andrew Bruce & Peter Gedeck.

[4] Song YY, Lu Y. Decision tree methods: applications for classification and prediction. Shanghai Arch Psychiatry. 2015 Apr 25;27(2):130-5. doi: 10.11919/j.issn.1002-0829.215044. PMID: 26120265; PMCID: PMC4466856

[5] Breiman, L. Random Forests. Machine Learning 45, 5–32 (2001). https://doi.org/10.1023/A:1010933404324

[6] Máquina de Soporte Vectorial (SVM). 2020 https://medium.com/@csarchiquerodriguez/maquina-de-soporte-vectorial-svm-92e9f1b1b1ac

[7] Understanding RNN and LSTM (SVM). Aditi Mittal Oct. 4 2019 https://aditimittal.medium.com/understanding-rnn-and-lstm-f7cdf6dfc14e