Acta de Correcciones al Proyecto de Grado
Ingeniería de Sistemas y Computación

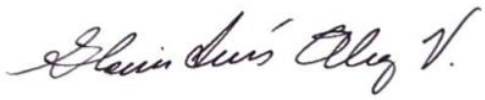
Fecha: 04/03/2024

Autores: Juan Pablo Giraldo Mosquera

Nombre del Proyecto de Grado: Diseño de un aplicativo para el análisis de sentimiento de reseñas de películas

Director: Dra. Gloria Inés Álvarez Vargas

Como indica el artículo 2.27 de las Directrices de Trabajo de Grado, he verificado que los estudiantes indicados arriba han implementado todas las correcciones que los Jurados del Proyecto de Grado definieron que se efectuaran, como consta en el Acta de Calificación correspondiente.



_____

Firma de Director(a) del Proyecto de Grado

Nota de Aceptación

Aprobado por el Comité de Trabajo de Grado en cumplimiento de los requisitos exigidos por la Pontificia Universidad Javeriana para optar el título de Ingeniero de Sistemas y Computación.

Dr. CAMILO ROCHA
Decano de la Facultad de Ingeniería

Dr. GERARDO MAURICIO SARRIA
Director Carrera Ingeniería Sistemas y Computación.

Dra. GLORIA INÉS ÁLVARES VARGAS
Director(a) Trabajo

Dr. GERARDO MAURICIO SARRIA
Jurado 1

MSc. MARIO JULIÁN MORA
Jurado 2

Pontificia Universidad Javeriana Cali Facultad de Ingeniería.
Ingeniería de Sistemas y Computación.
Trabajo de Grado.

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

Juan Pablo Giraldo Mosquera

Directora: Dra. Gloria Inés Álvarez Vargas

Febrero 12, 2024

Pontificia Universidad
JAVERIANA
Cali

Santiago de Cali, Febrero 12 de 2024

Señores
Pontificia Universidad Javeriana Cali
Director Carrera de Ingenieria de Sistemas y Computación
La ciudad

Cordial saludo,

Me permito informarle que el estudiante de Ingeniería de Sistemas y Computación Juan Pablo Giraldo Mosquera con código 8934332 trabajó bajo mi dirección en el Trabajo de Grado denominado “Diseño de un Aplicativo para el Análisis de Sentimiento de Reseñas de Películas” el cual está listo para ser sustentado.

Atentamente,

_____
Dra. Gloria Inés Álvarez Vargas

Santiago de Cali, Febrero 12 de 2024

Señores
Pontificia Universidad Javeriana Cali.
Dr. Gerardo Mauricio Sarria
Director Carrera de Ingeniería de Sistemas y Computación.
Cali.

Cordial Saludo.

Me permito presentar a su consideración el Trabajo de grado titulado “Diseño de un aplicativo para el análisis de sentimiento de reseñas de películas” con el fin de cumplir con los requisitos exigidos por la Universidad para llevar a cabo el proyecto de grado y posteriormente optar al título de Ingeniero de Sistemas y Computación.

Al firmar aquí, doy fe que entiendo y conozco las directrices para la presentación de trabajos de grado de la Facultad de Ingeniería aprobadas el 26 de Noviembre de 2009, donde se establecen los plazos y normas para el desarrollo del anteproyecto y del trabajo de grado.

Atentamente,

Juan Pablo Giraldo Mosquera
Código: 8934332

Resumen

Las plataformas en línea han permitido que los individuos puedan compartir sus experiencias y opiniones sobre todo tipo de productos y servicios del mercado a nivel global. Esta información se convierte en un recurso de gran valor para las empresas siempre que éstas puedan identificar el sentimiento de sus consumidores sobre los productos y/o servicios que ofrecen. Es por eso que esta tarea representa uno de los intereses principales en la actualidad a nivel comercial.

En el presente trabajo de grado se evaluará el desempeño de tres modelos de clasificación para la identificación del sentimiento de textos extraídos de reseñas de películas de la plataforma IMDB. En donde se seleccionará el más prometedor de ellos para la posterior implementación de una aplicación que permita a los usuarios aprovechar dicho modelo.

Palabras Clave: Análisis del sentimiento, aprendizaje automático, plataformas en línea, opiniones, reseñas de películas, redes neuronales, modelos de clasificación automática.

Índice General

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 implementarán tres 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.

Capítulo 1

Descripción del Problema

1.1. Planteamiento del Problema

La facilidad de acceso y libertad que ofrece Internet a las comunidades de hoy en día ha permitido que la opinión de cada individuo pase a ser un recurso de gran valor para diversos nichos de mercado. Estas opiniones están generalmente representadas por textos de mensajes de usuarios en distintas plataformas, en donde la problemática que surge es la clasificación del sentimiento o "tono" de dichos textos.

Esta tarea de clasificación, comúnmente conocida como análisis de sentimiento, aporta a las empresas la percepción que se tiene de sus productos y/o servicios por parte de sus consumidores. Ahora bien, delimitando el dominio a un caso particular, observemos una sala de cine o una plataforma de streaming las cuales podrían usar el análisis de sentimiento sobre diversas reseñas de películas con el fin de seleccionar, en su lista de emisión, sólo aquellas que tengan una mayor cantidad de críticas positivas; así mismo, los espectadores tendrían un interés por conocer qué películas tienen una mejor acogida por los críticos con el fin de decidir si verla o no. Los datos de entrada para este proceso de análisis constan de textos de moderada longitud que podrían tener la siguiente forma:

"IT IS A PIECE OF CRAP! not funny at all. during the whole movie nothing ever happens. i almost fell asleep, which in my case happens only if a movie is rally bad. (that is why it didn't get 1 (awful) out of 10 but 2).don't be fooled, like i was, by first review. a waste of money and your time! spend it on other stuff. at this point i'm finished with my review but i have to fill in at least ten lines of text so i will go on...."

Así mismo, la salida del proceso es una etiqueta del sentimiento que expresa esa porción de texto pudiendo clasificarse en una de dos clases: positiva o negativa. De este modo, el problema computacional que se quiere abordar con este proyecto es el de automatizar el análisis de sentimiento y permitir gestionar las películas de acuerdo a las reseñas.

1.1.1. Formulación

¿Cómo realizar un análisis de sentimiento de reseñas de películas haciendo uso de técnicas de aprendizaje automático e inteligencia artificial?

1.1.2. Sistematización

- ¿Qué características debe cumplir el conjunto de los datos de entrada?
- ¿Cómo construir tres modelos de predicción que hagan uso de técnicas distintas?
- ¿Cómo se medirá el desempeño de los modelos desarrollados?
- ¿Cómo se seleccionará el modelo más prometedor?
- ¿Cómo permitir que los espectadores puedan aprovechar el modelo más prometedor para elegir las películas que van a ver?

1.2. Objetivos

1.2.1. Objetivo General

Construir un modelo que permita identificar el sentimiento asociado a reseñas de películas de IMDB.

1.2.2. Objetivos Específicos

- Definir el conjunto de datos de entrada y sus características.
- Construir los modelos correspondientes a las técnicas seleccionadas para la tarea de clasificación.
- Evaluar el desempeño de los modelos.
- Seleccionar el modelo más prometedor de acuerdo a su desempeño.
- Implementar una aplicación que permita la interacción entre los usuarios/espectadores y el modelo elegido.

1.3. Justificación

El análisis de sentimiento es una tarea de gran interés para las empresas de hoy en día. Lo anterior se debe a que en esta era digital las redes sociales, las reseñas, los foros y en general cualquier plataforma que permita socializar las experiencias de los usuarios con distintos productos y/o servicios se han vuelto la principal referencia de los consumidores. Así mismo, la fácil accesibilidad que tienen estas plataformas se vuelve un arma de doble filo debido al crecimiento exponencial del volumen de los datos generados por los mismos usuarios, los cuales llegan a un punto en que la realización de un análisis de sentimiento de manera manual se vuelve prácticamente imposible, por lo que surge la necesidad de buscar una solución a este problema de ejecución mediante la automatización de la tarea.

Ahora bien, centrándonos en nuestro dominio de interés, específicamente para las plataformas de streaming y las salas de cine resulta de gran importancia conocer la acogida que tienen las películas que emiten o piensan emitir de cara a estipular el tiempo que estas estarán disponibles o si es factible si quiera añadirlas a su lista de emisión. Esto mediante las reseñas que se pueden obtener en distintos foros o plataformas de ranking de películas como lo es IMDB, reddit o en general cualquier red social en la que se publiquen opiniones de las películas en cuestión. Así mismo, para algunos espectadores es importante conocer la crítica que tiene una película determinada, pues esto usualmente es la base de su criterio para decidir si verla o no.

Desde la computación podemos explorar una comparación de desempeño entre el uso de clasificadores clásicos del aprendizaje automático frente a técnicas más avanzadas como redes neuronales. De este modo podremos seleccionar el modelo más prometedor para su posterior integración a la implementación de una aplicación que permita a los usuarios aprovecharlo al momento de gestionar las películas que desean ver.

1.4. Alcances y Limitaciones

- Con el proceso de investigación se buscará seleccionar dos técnicas que hagan uso de alguno de los clasificadores clásicos del aprendizaje automático para contrastar su desempeño frente a una técnica más avanzada (deep learning).

- El dominio de interés de esta investigación se limita al de reseñas de películas extraídas de diversas páginas web de opinión, foros y redes sociales; teniendo como principal fuente a www.imdb.com. Consecuentemente, el modelo de clasificación se limita a textos de un promedio de 200 palabras de longitud en idioma inglés.

- Para la evaluación del desempeño de los modelos construidos se usarán las métricas de precisión, exactitud, exhaustividad y puntaje f1.

- La calidad de los modelos así como las conclusiones a las que llegue respecto al contraste de técnicas se ceñirá única y exclusivamente a las métricas mencionadas en el punto anterior.

- Con las métricas de evaluación de desempeño se seleccionará uno de los tres modelos como el más prometedor.

1.4.1. Entregables

Al finalizar el proyecto se contará con los siguientes entregables:

- Un documento de tesis.

- Implementación de una aplicación que permita a los espectadores aprovechar el modelo de clasificación.

- Un corpus compuesto de reseñas de películas. Este se define como el conjunto procesado de los textos. Este entregable solo aplica si se hacen modificaciones propias en el conjunto, en ningún caso se incluirá si no se realizan modificaciones significativas en los conjuntos extraídos de la web.

Capítulo 2

Desarrollo del Proyecto

2.1. Marco de Referencia

En esta sección se expondrán los conceptos básicos fundamentales que están involucrados tanto en el análisis de sentimiento como en el aprendizaje automático, sus técnicas clásicas y el aprendizaje profundo.

2.1.1. Análisis de sentimiento

Las opiniones, al igual que sus sinónimos tales como sentimientos, evaluaciones, actitudes y emociones de los individuos son fundamentales para casi todas las actividades humanas, pues la toma de decisiones y la percepción de la realidad de las personas y las organizaciones se basa principalmente en la forma en la que los otros ven y evalúan el mundo. En la actualidad, el grado de influencia de estas opiniones es el tema central del análisis de sentimiento, el cual es el campo de estudio que analiza las opiniones a través de productos, servicios, organizaciones, individuos, eventos, y sus atributos [1].

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 humano, por lo que desde la computación se ofrece una solución haciendo uso de las técnicas del aprendizaje automático.

2.1.2. Aprendizaje automático

La computación nace como respuesta a la necesidad de automatizar tareas que se han tornado en laborales imposibles de realizar para una persona, o inclusive, para un numeroso grupo de ellas debido al volumen desmesurado de los datos a procesar. Esta necesidad es atendida

desde la computación por medio de la implementación de algoritmos, cuya función consiste en la transformación de "datos en bruto" (o datos de entrada) a un conjunto de datos finales deseados a través de una serie de pasos traducidos en instrucciones ejecutables por una máquina que simularán las acciones que tendría que hacer una persona para completar dicha transformación por su propia mano. Finalmente, un algoritmo ofrece una solución trivial para el problema de automatización de tareas repetitivas con pasos bien definidos que se mantienen constantes sin importar las características de los datos de entrada.

Para algunas tareas, sin embargo, no es posible construir un algoritmo constante que permita procesar cualquier entrada. Por ejemplo, para la tarea de clasificar correos electrónicos como legítimos o "spam" (anglicismo que refiere principalmente a correos con información publicitaria) únicamente se conocen los datos de entrada los cuales se componen principalmente de documentos que, en el caso más simple, contienen determinado número de caracteres, y los datos de salida o finales que consisten en un conjunto de etiquetas con dos posibles valores (si/no indicando respectivamente si el correo es spam o no), sin embargo, no existe una forma de plasmar los pasos necesarios para la realización de esa tarea de clasificación puesto que requiere de experiencia previa o de datos de ejemplo para poder diferenciar un tipo de correo de otro [2]. De este modo, el aprendizaje automático es el campo de estudio que ofrece diversas técnicas que nos permiten entrenar un algoritmo para que tenga la capacidad de realizar este tipo de tareas que requieren de la experiencia previa para su correcta realización.

2.1.2.1. Técnicas del Aprendizaje Automático

El aprendizaje automático utiliza teoría de probabilidad y estadística para la construcción de modelos de inferencia que permitan predecir eventos futuros a partir de una muestra de datos previos. Esto mediante un proceso compuesto de una fase de entrenamiento que busca "enseñar" al algoritmo a realizar esta predicción y una fase de validación y prueba que consiste en la verificación de la efectividad y precisión de la predicción del modelo [2]. Sus modelos o clasificadores clásicos se encuentran distribuidos principalmente en dos clases que

- **Aprendizaje Supervisado:** Dentro de esta clasificación, la fase de entrenamiento consta de un histórico de los datos previamente etiquetado (es decir, se conocen tanto los datos de entrada como los de salida) cuya intención es enseñar al algoritmo a reconocer cada clase de la clasificación "observando" cómo se han etiquetado en el pasado [3]. Existen numerosos algoritmos que usan este tipo de aprendizaje. A continuación, se exponen cuatro de ellos que competen a este proyecto de grado:

  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.

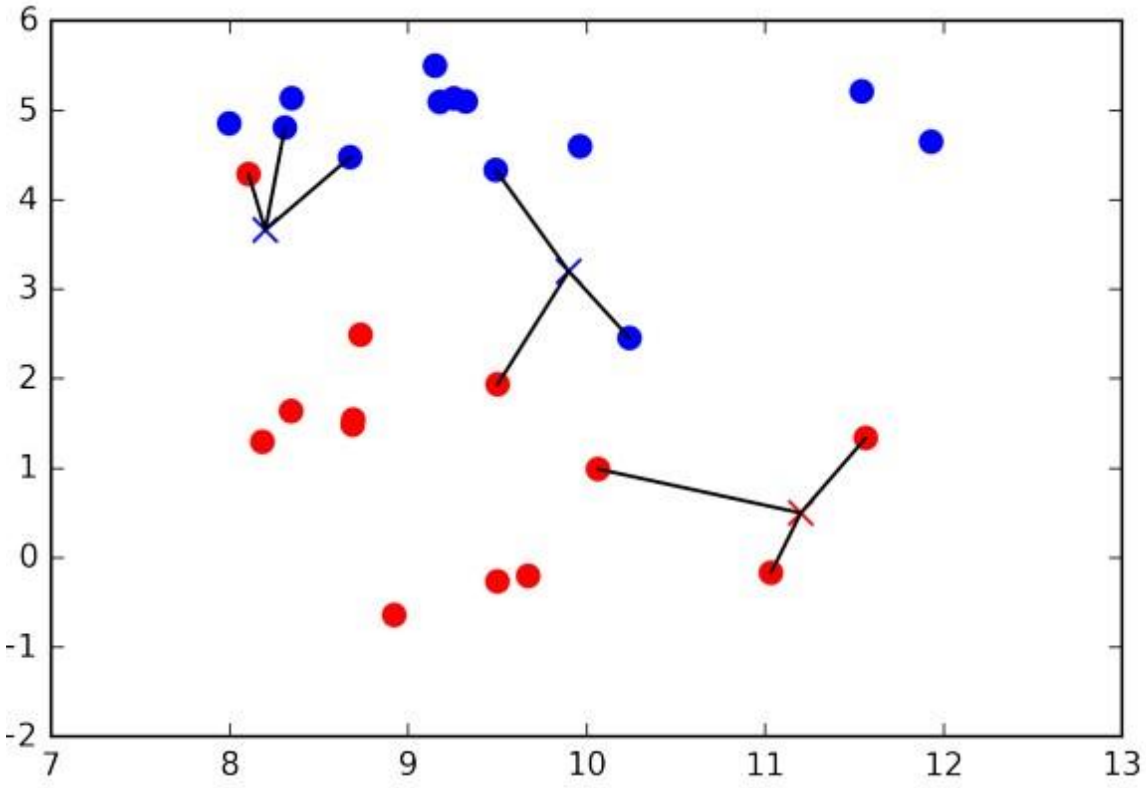
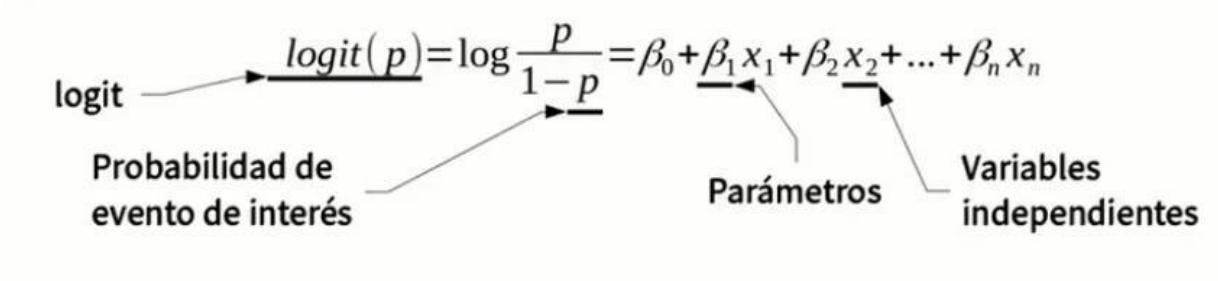


Figura 1: K-Nearest Neighbors [4].

Como se observa en la figura, la predicción inicia localizando determinado número de datos previos más cercanos al entrante (vecinos) y este último se clasifica dentro del grupo al que pertenecen la mayoría de sus vecinos. Los datos entrantes se representan con una "X" mientras que los previos son los puntos pertenecientes a cada grupo distinguido por colores. Los parámetros relevantes en este caso son el número de vecinos a tener cuenta para la predicción; y la forma en que se calcula la distancia entre ellos y el dato 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 un rango de ± ∞. La clasificación finalmente, está representada con una curva en donde se retorna 1 si el dominio tiende a infinito positivo o cero si tiende a infinito negativo, por lo que claramente este clasificador solo efectivo en problemas de clasificación binaria. [5]



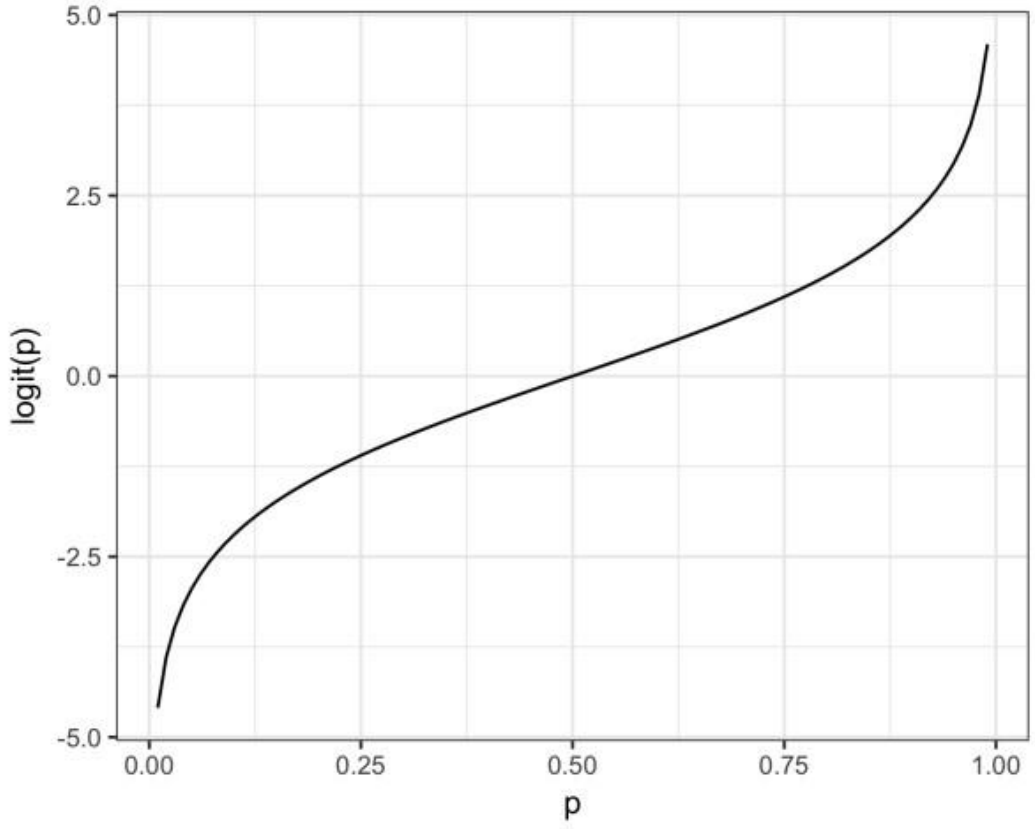


Figura 2: Función Logit [5]

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. Las ramas conectan los nodos resultantes de una decisión; un nodo que origina nuevos eventos (o nodos) se le conoce como nodo padre mientras que los nuevos nodos serán sus hijos, luego las ramas son las que reflejan esta relación. [6] Dentro del algoritmo se conocen tres tipos de operación:

- **Splitting:** Mediante las variables relacionadas con el atributo que se desea predecir se toman decisiones o se responden preguntas sobre sus valores. Esta operación es la que crea nuevos nodos dentro del árbol.
- **Stopping:** Con el fin de evitar el sobre entrenamiento y escalar el tamaño del árbol se deben crear condiciones de parada parametrizándolas con valores que acoten el tamaño del árbol. Por ejemplo, altura máxima; y el número mínimo y máximo de hojas. Esta operación es la que logra que el árbol converja a un resultado final sobre el atributo de interés.
- **Pruning:** En algunas situaciones, las condiciones de parada pueden ser poco efectivas para acotar de manera eficiente la construcción del árbol de decisión. Luego, es necesaria esta tercera operación de poda o "Pruning" la cual consiste en la eliminación de los nodos que aporten menos información relevante al atributo de interés y solo mantener aquellos más que otorguen un resultado óptimo.

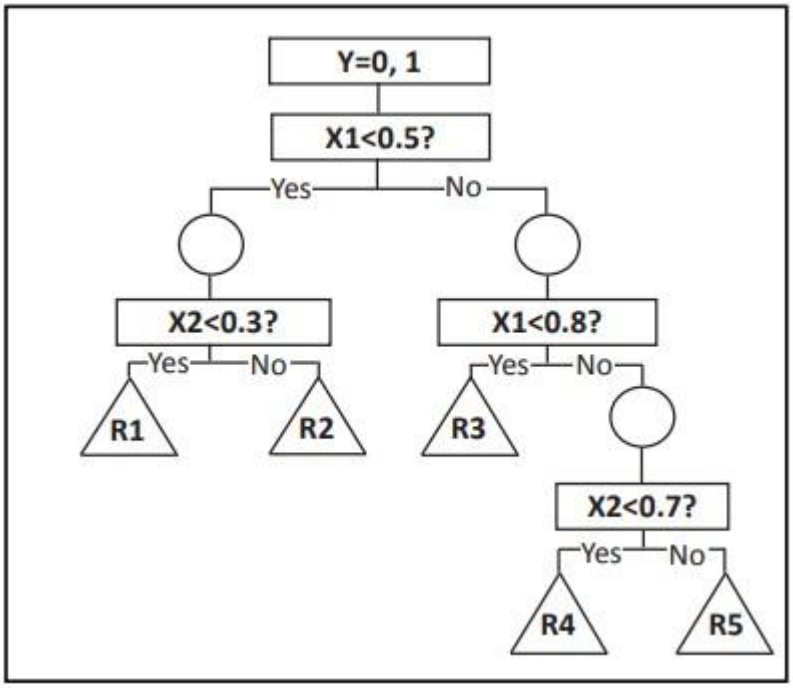


Figura 3: Decision Tree [6]

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]

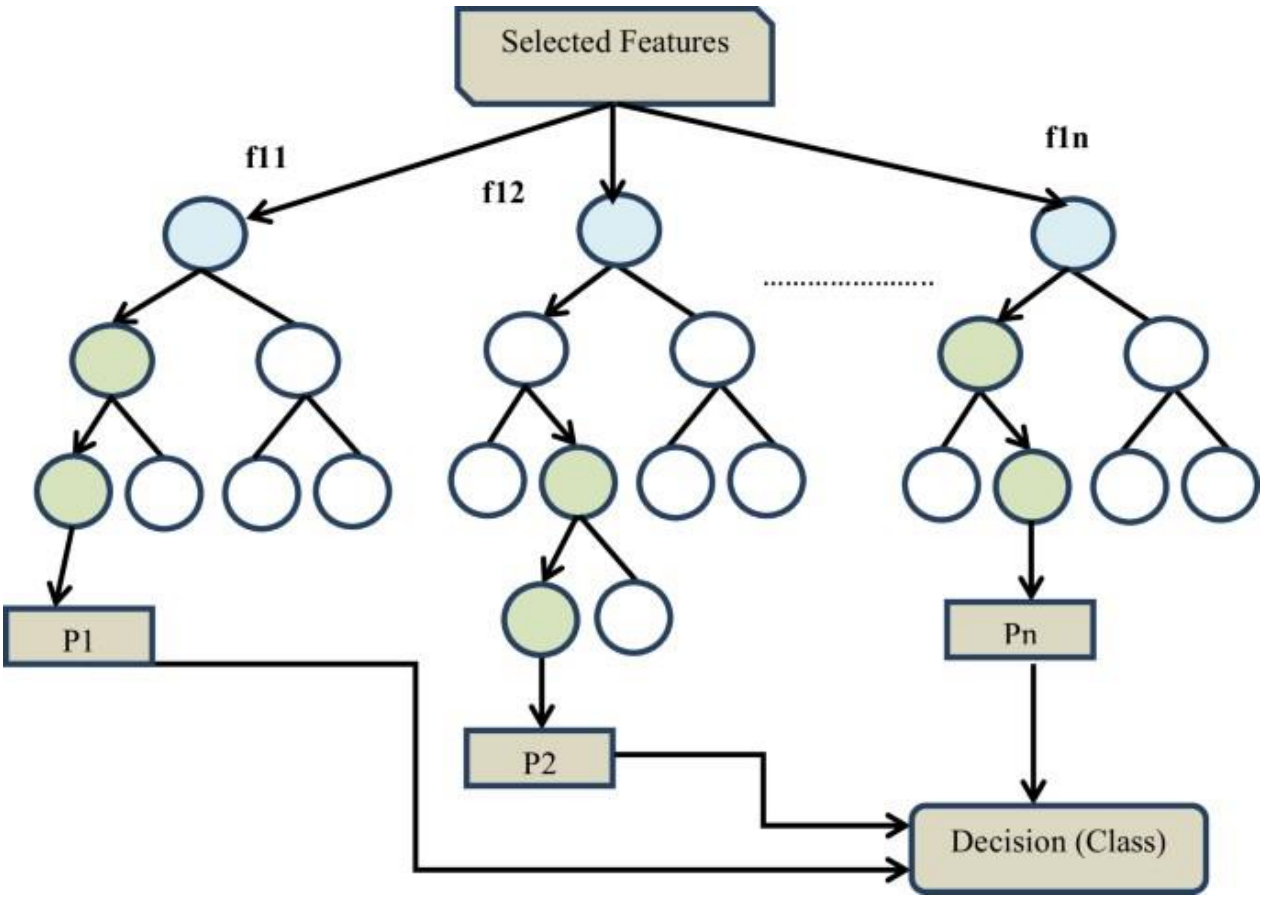


Figura 4: Random Forest [8]

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]

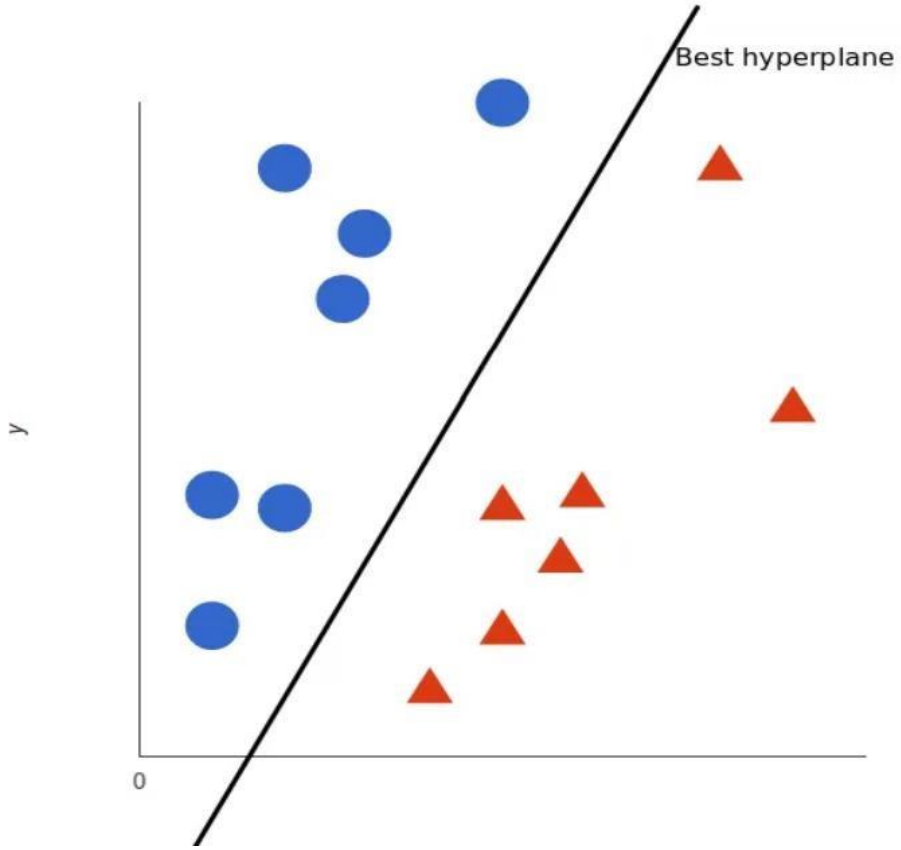


Figura 5: Máquina de Vectores de Soporte Binaria [9]

2.1.3. Aprendizaje profundo y Redes Neuronales Artificiales

Como bien se define en investigaciones recientes (Qin, 2021), el aprendizaje profundo se basa principalmente en las redes neuronales artificiales (ANNs), las cuales comprenden una colección de nodos conectados entre sí. La red más simple se compone de un solo nodo o neurón, también llamado perceptrón, el cual combina múltiples entradas, ejecuta transformaciones no lineales, y retorna un valor escalar. Matemáticamente, un perceptrón puede ser representado por una función de activación “g” que toma un vector “x” de entrada y devuelve un escalar “y” como se aprecia en las siguientes figuras [10].

$$y = g(\sum_{j=1}^{d} w_j x_j + b),$$

Figura 6: Representación Matemática de un Perceptrón. [10]

$$g(z) = \sigma(z) = \frac{1}{1 + e^{-z}}.$$

Figura 7: Función de Activación. [10]

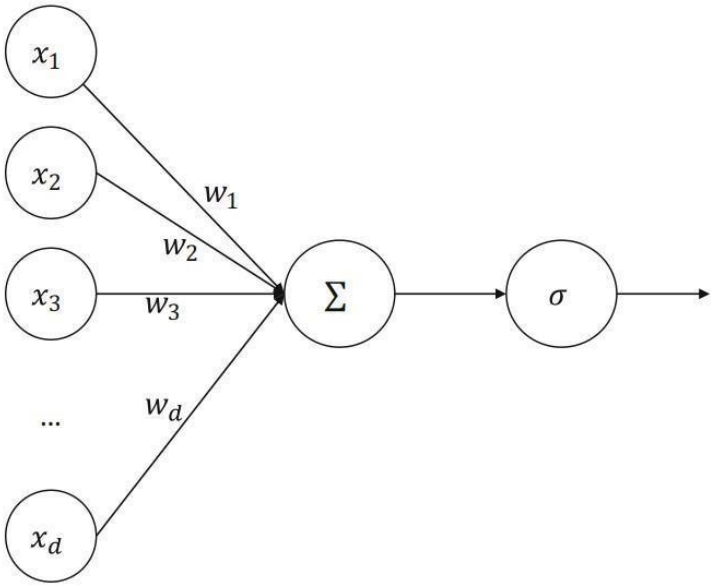


Figura 8: Red Neuronal más simple, originalmente inspirada en los sistemas biológicos humanos [10].

En donde el escalar resultante "y" representa la probabilidad de que la muestra de entrada "x" pertenezca a una clase y "1-y" representa la probabilidad de que la muestra "x" pertenezca a otra clase dentro de un problema de clasificación binario [10].

2.1.4. 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] Los tipos de compuertas se exponen a continuación:

- **Input Gate – Compuerta de Entrada:** Compuesta por una función de activación sigmoide, decide que valores mantienen o desechan de la red asignando un valor entre 0 y 1 según se evalúe. Para los datos que se mantienen en la red, una función de activación tangente asigna un peso o valor de relevancia entre [-1,1]. (En el caso de una red recurrente básica, únicamente existe la función de activación tangente que asigna el nivel de importancia). Su representación matemática se aprecia en las respectivas fórmulas:

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

- **Forget Gate – Compuerta del Olvido:** Esta compuerta valida el estado anterior h(t-1) y el contenido de la entrada actual (Xt) decidiendo mediante una función de activación sigmoide la secuencia de entrada que se mantendrá en la red devolviendo un número entre 0 (omitir los datos) y 1 (mantener los datos) para cada número en la celda de estado Ct-1. A continuación la expresión matemática de esta compuerta:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

- **Output Gate – Compuerta de Salida:** Cumple una función similar a la puerta de entrada en cuanto a las funciones de activación utilizadas añadiendo un paso extra que consiste en calcular el producto entre el resultado de la función sigmoide con la asignación del nivel de relevancia en función de calcular el nuevo estado h[t] de la red.

$$o_t = \sigma\left(W_o \; [\, h_{t-1}, x_t] \; + \; b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

En la figura a continuación se ilustra la arquitectura completa de una celula LSTM que describe el funcionamiento anteriormente mencionado:

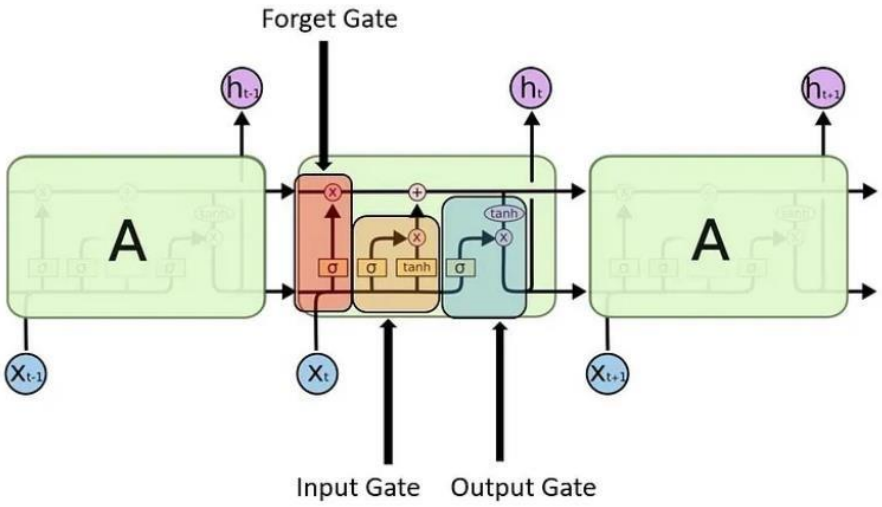


Figura 10: Celda LSTM [11]

2.1.6 Procesamiento del Lenguaje Natural (NLP)

El NLP es un subcampo de la ciencia de la computación cuya función radica en permitir a un computador entender el lenguaje de una manera "natural", es decir, como los humanos lo comprenden. Generalmente, esto se traduce a tareas como el entendimiento del sentimiento de un texto, reconocimiento de discursos, y la generación de respuestas a preguntas expresadas en este tipo de lenguaje. Algunos ejemplos de la aplicación del NLP se observan en los chatbots que se encargan del servicio al cliente, auto correctores gramaticales de los dispositivos móviles y los asistentes inteligentes como Cortana o Siri que también se encuentran en los anteriormente mencionados [12].

2.1.6. Reseña Fílmica

La reseña es un tipo de texto, generalmente de corta extensión, que se hace sobre una obra literaria, de arte o científica y se publica en un periódico o en una revista [13]. Para el dominio de este proyecto, este recurso textual comprende, específicamente, un propósito híbrido: presentar una síntesis de la película y argumentar una valoración crítica con base en el análisis de la misma [14].

2.1.5. Antecedentes de investigación

El dominio de la tarea de clasificación de análisis de sentimiento que se tomará como foco principal de este proyecto se concentra, principalmente, en la extracción de la opinión de los usuarios expresada a través de la publicación de reseñas de películas en la web. Lógicamente, como los usuarios construyen las reseñas a través del lenguaje natural, la identificación del

sentimiento de las mismas se enfrenta a la dificultad de poder procesar los datos de entrada mediante un algoritmo.

Soubraylu y Rajalakshmi [15], a diferencia de la metodología que se llevará a cabo en este trabajo, abordan esta problemática mediante un enfoque de procesamiento de los textos a nivel de frases en lugar de realizar transformaciones de secciones unitarias del texto (palabras) como un Word2Vec. Su implementación comprende un modelo de Red Neuronal Convolucional Bidireccional Recurrente (CBRNN) que combina una Red Neuronal Convolucional de dos capas (CNN) y una Unidad Bidireccional Recurrente Cerrada (BGRU). El modelo CBRNN propuesto extrae las características de los textos a nivel de frases, además de las características cronológicas a largo plazo que distinguen a cada clase. De este modo, obtienen una mejoría en la precisión del modelo a comparación de trabajos previos que usan el enfoque a nivel de palabras, sin embargo, acotan que esta implementación lleva consigo un costo computacional significativo.

Por otro lado, Qaisar [16] estipula que ambos enfoques se enfrentan dos desafíos claves que se siguen tratando en las investigaciones de hoy en día. El método de transformación de los textos a vectores de palabras entra en conflicto con palabras ambiguas que poseen múltiples significativos dentro de un mismo contexto, mientras que el enfoque a nivel de frases resulta en una implementación de alto riesgo puesto que estas pueden no comunicar nada en específico perdiendo relevancia al momento de entrenar el modelo. En su implementación, sin embargo, se enfoca en el problema del costo temporal y espacial mediante un modelo *Long-Short Term Memory (LSTM)* para la clasificación del sentimiento de reseñas de películas extraídas de la web IMDB siguiendo una estrategia de partición del conjunto de datos de entrada para las fases en entrenamiento, validación y prueba, similar a la que se llevará a cabo en el presente proyecto, con la cual se obtuvo un mejor rendimiento a nivel computacional. En nuestro caso, si bien el costo computacional es uno de los factores que se evaluará dentro de los criterios de desempeño de los modelos que se quieren construir, éste no compone el foco principal para la selección del más prometedor de ellos que posteriormente se utilizará para la implementación de la aplicación.

Siguiendo esta misma línea de prioridades en las métricas de evaluación de desempeño de los modelos, Novelty Octaviani [17] se enfoca en la maximización del desempeño del clasificador KNearest Neighbors mediante el uso de características del método selección information gain el cual está presente en el clasificador de árboles de decisión.

Finalmente; Dashtipour, Gogate, Adeel, Larijani y Hussain [18] corroboran el trabajo de Qaisar al implementar su mismo modelo y obtener un mejor desempeño tras comparar los resultados con los modelos de perceptrón multicapa (MLP), Máquina de vectores de soporte (SVM), regresión logística y algoritmos de redes computacionales convolucionales (CNN). Cabe destacar, que nuestro objetivo no se compromete en la continuación de la labor de estos autores, pues no se busca el enaltecimiento de un clasificador en particular, sino más bien seleccionar el modelo más prometedor para la implementación de una aplicación.

2.2. Metodología

2.2.1. Tipo de Estudio

Este proyecto comprende un tipo de estudio experimental, en donde su

desarrollo consta del preprocesamiento de los datos de entrada, la selección de las técnicas que usarán los diferentes modelos, la estimación de los parámetros de las mismas, la ejecución de los modelos y la posterior medición del desempeño de los mismos a través de las métricas de precisión, exhaustividad, exactitud y puntaje f1.

2.2.2. Actividades

Las actividades a desarrollar para el cumplimiento de los objetivos se enumeran a continuación desglosando todos los pasos que se llevarán a cabo para su ejecución.

2.2.2.1. Definir el conjunto de datos de entrada y sus características.

Este objetivo comprende la primera parte de la implementación del proyecto. Para su cumplimiento se llevarán a cabo los siguientes pasos:

- Recolectar un conjunto de reseñas de películas disponible en internet. Esto no comprende un proceso de construcción manual propia, en cualquier caso, se extraen conjuntos previamente construidos que estén disponibles en internet.

- En caso de necesitar un mayor volumen, se concatenarán distintos conjuntos con el fin de enriquecer el proceso de entrenamiento y posterior validación de los modelos.

- Preparación y preprocesamiento de los textos.

- Realizar una conversión “Word2vec” la cual consiste en el proceso de conversión de los textos a vectores numéricos.

2.2.2.2. Construir los modelos correspondientes a las técnicas seleccionadas para la tarea de clasificación.

- Investigar sobre los clasificadores clásicos de aprendizaje automático y las técnicas de aprendizaje profundo más adecuadas para esta tarea de clasificación.

- Seleccionar dos técnicas que hagan uso de clasificadores clásicos y una de aprendizaje profundo para la construcción de un total de tres modelos de clasificación.

- Diseñar experimentos que permitan evaluar el desempeño de los modelos para esta tarea de clasificación.

- Seleccionar el conjunto total de datos de manera balanceada para su uso en las fases de entrenamiento, validación y prueba del experimento.

- Estimar los parámetros de cada técnica en función del mejor desempeño.

- Ejecutar los experimentos diseñados para los tres modelos de clasificación usando la porción del conjunto de datos destinado en cada fase.

2.2.2.3. Evaluar el desempeño de los modelos.

- Medir el desempeño de los modelos mediante el cálculo de las métricas de precisión, exhaustividad, exactitud y puntaje f1.

- Definir los criterios de evaluación de desempeño estableciendo rangos de valores ideales que se espera obtener en las métricas

- Realizar modificaciones en los experimentos con el fin de maximizar el desempeño de cada uno de los modelos en caso de ser posible, e iterar nuevamente para reevaluar el desempeño.

2.2.2.4. Seleccionar el modelo más prometedor de acuerdo a su desempeño.

- Comparar el desempeño de los tres modelos en función de los valores obtenidos en sus métricas

- Seleccionar el modelo más prometedor de acuerdo al criterio definido. Se entiende por modelo más prometedor a aquel que logre un mayor puntaje en sus métricas.

2.2.2.5. Implementar una aplicación que permita la interacción entre los usuarios/espectadores y el modelo elegido.

- Definir los requerimientos funcionales y no funcionales del usuario objetivo (en este caso la aplicación estará orientada principalmente a los espectadores de las salas de cine y a los usuarios de las plataformas de streaming).

- Establecer las tecnologías se usarán para la implementación de la aplicación.

- Diseñar la arquitectura de la aplicación de acuerdo a los requerimientos establecidos.

- Implementar la aplicación siguiendo el diseño propuesto. Cabe destacar que la implementación incluirá únicamente el modelo seleccionado como el “más prometedor”.

Capítulo 3

Preprocesamiento de los Datos

3.1. Construcción del Corpus

El corpus del proyecto se compone de dos conjuntos de reseñas de películas de la web IMDB, concatenados mediante la librería pandas cuyos únicos atributos son el texto de la reseña y la etiqueta de sentimiento (positivo o negativo). A continuación, se expone una incidencia de cada clase para ejemplificar los tipos de texto que se encuentran en el corpus:

- **Incidencia Reseña Negativa.**

"I grew up (b. 1965) watching and loving the Thunderbirds. All my mates at school watched. We played "Thunderbirds" before school, during lunch and after school. We all wanted to be Virgil or Scott. No one wanted to be Alan. Counting down from 5 became an art form. I took my children to see the movie hoping they would get a glimpse of what I loved as a child. How bitterly disappointing. The only high point was the snappy theme tune. Not that it could compare with the original score of the Thunderbirds. Thankfully early Saturday mornings one television channel still plays reruns of the series Gerry Anderson and his wife created. Jonatha Frakes should hand in his directors chair, his version was completely hopeless. A waste of film. Utter rubbish. A CGI remake may be acceptable but replacing marionettes with Homo sapiens subsp. sapiens was a huge error of judgment."

- **Incidencia Reseña Positiva.**

"I love this movie, one of my all time favorites. Ann Blythe as Sally O'Moyne is sweet and trouble-free. She believes that praying to Saint Anne will solve all her and her friends troubles. The sub-plot of the dastardly bad man to get her father's property is funny and clever. Her brothers are what kind of brothers any girl would love to have. Also, look for "Aunt Bee" as her mother, a strong Irish woman who won't leave her house that she brought her family up in. They don't make them like this anymore, that's for sure."

Siguiendo con el procesamiento del corpus, se reemplazan las clases expresadas en texto como "positivo" y "negativo" por los números 1 y 0 respectivamente. Así mismo, se ejecuta un preprocesamiento del corpus como sigue:

- **Limpieza Inicial del Texto:** Eliminación de todos los caracteres no-alfanuméricos y etiquetas html que presentan algunas incidencias del corpus.

- **Eliminación de "Stopwords":** Eliminación de todas las palabras que por sí solas no aportan información necesaria para el proceso de clasificación como adverbios, conjunciones, artículos y preposiciones.

- **Stemming o Lematización:** Se eliminan todas las conjugaciones del texto llevando las palabras a su raíz singular base. Por ejemplo "Watching" es reducido a "Watch".

En la **Tabla 1** se expone la distribución de las clases del corpus completo. Como se aprecia en la cardinalidad de cada clase, se trata de un conjunto de datos lo suficientemente balanceado (prácticamente el 50% del corpus para cada clase) por lo que se decide no hacer un balance adicional.

Clase	Positivo	Negativo
Número de Reg.	44981	45019

Tabla 1: Cardinalidad de las Clases

Seguidamente se exponen las características del corpus resultante y el cambio de longitud que sufre tras el procesamiento descrito anteriormente. El conjunto de datos "Inicial" comprende la concatenación de ambos conjuntos de reseñas sin ningún procesamiento adicional mientras que el "Procesado" representa el conjunto tras la "Limpieza Inicial", la eliminación de Stopwords y la aplicación del proceso de Stemming.

Conjunto de Datos	Inicial	Procesado
Longitud Promedio (Caracteres)	1079	647
Longitud Promedio (Palabras)	231	129

Tabla 2: Cardinalidad de las Clases

3.2. Vectorización

Tras la construcción y el preprocesamiento del corpus se realiza la vectorización de los textos resultantes mediante el método *TF-IDF Vectorizer (Term Frequency-Inverse Document Frequency Vectorizer)* el cual construye vectores numéricos que representan el texto, otorgando un peso a cada palabra en función de su frecuencia, de modo que las palabras, o "tokens", que aparecen un mayor número de veces en el corpus poseen un peso o valor TF-IDF menor (es decir que son menos relevantes para la clasificación). Para mantener una dimensión fija se configura un máximo de características a considerar de 10.000 (max_features) debido a la longitud máxima que se encuentra en el corpus y un filtro max_df = 0.5 en donde las palabras con una frecuencia mayor a este parámetro se ignoran para el proceso de vectorización.

Capítulo 4

Construcción de los Modelos

4.1. Selección de los Modelos de Aprendizaje Automático

Considerando que se cuenta con un corpus previamente etiquetado, se seleccionaron cuatro modelos de clasificación de **aprendizaje supervisado** a tomar en consideración para la evaluación de desempeño teniendo cuenta los más usados dentro de la literatura de trabajos relacionads. Estos se listan a continuación:

- *Logistic Regression*
- *K-nearest Neighbors*
- *Random Forest*
- *Support Vector Machine*

Se realiza la evaluación de los modelos con sus parámetros por defecto para determinar el punto de partida del desempeño. El criterio de evaluación se limita al cálculo de los indicadores de f1 score, precisión, accuracy y recall. En la **Tabla 3** se exponen los valores de los indicadores de desempeño para los parámetros por defecto.

HYPERPARAMETER SEARCH & MODEL EVAL.	
Classifier	**Performance**
Logistic Regression	**Precision**: 0.8443
	Recall: 0.8922
	Accuracy: 0.8809
	F1 Score: 0.9005
Knearest Neighbors	**Precision**: 0.7531
	Recall: 0.8043
	Accuracy: 0.7900
	F1 Score: 0.7823
Random Forest	**Precision:** 0.8826
	Recall: 0.8956
	Accuracy: 0.8745

	F1 Score: 0.9136
Support Vector Machine	**Precision**: 0.8953
	Recall: 0.9133
	Accuracy: 0.9033
	F1 Score: 0.9042

Tabla 3: Desempeño de los modelos con los valores por defecto de los parámetros

4.1.1. Búsqueda de Hiper-parámetros y Evaluación de Clasificadores

Como se mencionó anteriormente, se acota el campo de selección de los modelos a aquellos clasificados dentro de la categoría de aprendizaje supervisado por las características del corpus. Así mismo, se realiza una búsqueda de los parámetros más óptimos usando el método **GridSearch** sobre determinados parámetros en cada modelo. En la **Tabla 4** encontramos por columnas los modelos evaluar, los parámetros considerados y sus respectivos rangos de valores a probar, así como las configuraciones más óptimas encontradas respectivamente:

CLASSIFIER	PARAM GRID	BEST PARAMS
Logistic Regression	C = [0.001, 0.01, 0.1, 1, 10, 100] solver = ['lbfgs', 'liblinear', 'newton-cg', 'saga'] penalty = ['none','l1', 'l2']	Best Penalty: l1 Best Solver: liblinear Best C: 1
K-nearest Neighbors	n_neighbors': [5, 7, 10, 15], 'weights': ['uniform', 'distance']	Best N. Neighbors: 10 Best Weights: uniform
Random Forest	n_estimators': [25, 50, 75, 100, 125, 150, 175, 200] 'max_depth': [10, 20, 30]	**Best N Estimators: 200** **Best N. Max Depth: 30**
Support Vector Machine	C': [0.1, 1, 10], 'gamma': [0.1, 1, 10], 'kernel': ['linear', 'rbf', 'sigmoid']	Best C: 1 **Best Gamma: 0.1** Best Kernel: rbf

Tabla 4: Resultados de la primera búsqueda de parámetros óptimos

En esta búsqueda inicial de los parámetros más óptimos notamos resultados en los extremos de los rangos de valores para algunos parámetros como se observan resaltados en rojo en la tabla anterior. Dado lo anterior, se decide realizar una segunda búsqueda extendiendo el rango en dirección a los extremos resultantes de la búsqueda inicial. En la **Tabla 5** se exponen los resultados de esta segunda búsqueda para los modelos “conflictivos”.

HYPERPARAMETER SEARCH & MODEL EVAL.		
Classifier	**Grid Search**	**Best Params.**
Logistic Regression	*[0.001, 0.01, 0.1, 1, 10, 100]*	***C=1***
	['lbfgs', 'liblinear', 'newton-cg', 'saga']	***solver=liblinear***
	['none','l1', 'l2']	*penalty=l1*
Knearest Neighbors	*[5, 7, 10, 15]*	***n**_neighbors=10*
	['uniform', 'distance']	***weights**=uniform*
Random Forest	*[300, 400, 450, 500, 550, 600]*	***n_estimators**= 400*
	[50, 60, 70, 80, 90, 100]	***max_depth**=60*
Support Vector Machine	*[0.001, 0.01, 0.1, 1, 10, 100]*	***C**=1*
	[0.0001,0.001,0.01,0.1, 1]	***gamma**=0.1*
	['linear', 'rbf', 'sigmoid']	***kernel**=rbf*

Tabla 5: Resultados de la segunda búsqueda de parámetros óptimos

Como se aprecia en la tabla, los valores más óptimos para cada parámetro corresponden a valores intermedios en cada rango, lo que nos da certeza de que no hay un valor más óptimo que los seleccionados a considerar.

4.2. Selección del Modelo de Aprendizaje Profundo.

Se seleccionó una Red Recurrente LSTM como el modelo de aprendizaje profundo. Este modelo es ideal para problemas de clasificación en donde el contexto de toda la secuencia de entrada es relevante para el resultado de la predicción. En este caso, se requiere conocer todo el contexto del texto para poder reconocer el sentimiento del mismo y reducir el fallo a causa de premisas satíricas o sarcásticas.

4.2.1. Búsqueda de Hiper-parámetros y Evaluación del Método Profundo

En la siguiente tabla se representan los experimentos realizados con sus respectivas variaciones en los valores de cada parámetro y en la arquitectura misma del modelo, aplicadas basándose en las técnicas comúnmente usadas en trabajos relacionados como mantener las unidades de capa en potencias de dos para facilitar las operaciones matriciales así como la variación del tamaño del batch y la inclusión del Early Stop lo cual permite controlar el sobre-entrenamiento del modelo basándose en la variación de

desempeño en cada época de la red neuronal. No se realiza una búsqueda exhaustiva debido a la inviabilidad relacionada con el tiempo de ejecución de cada experimento.

HYPERPARAMETER SEARCH & MODEL EVAL.			
Network Version	**Layers**	**Params**	**EarlyStopParms**
LSTM V.1 128 Units	LSTM - 128 Units Dense - 1 Unit Dropout (0.8)	*epochs=20*	
		batch_size=256	
		val_split=0.1	
LSTM V.2 128 Units	LSTM - 128 Units Dense - 128 Units Dense - 1 Unit Dropout (0.8)	*epochs=10*	
		batch_size=512	
		val_split=0.2	
LSTM V.3 128 Units	LSTM - 128 Units Dense - 64 Units Dense - 1 Unit Dropout (0.6)	*epochs=5*	
		batch_size=1024	
		val_split=0.2	
LSTM V.4 128 Units	LSTM - 128 Units Dense - 64 Units Dense - 1 Unit Dropout (0.5)	*epochs=50*	
		batch_size=1024	
		val_split=0.2	
LSTM V.5 128 Units	LSTM - 128 Units Dense - 64 Units Dense - 1 Unit Dropout (0.5)	*epochs=16/20(EarlyStop)*	patience = 3
		batch_size=1024	monitor=val_accuracy
		val_split=0.2	mode=max
BIDIRECTIONAL LSTM 128 Units	Bidirectional LSTM - 128 Units Dense - 64 Units Dense - 1 Unit Dropout (0.5)	*epochs=18/20(EarlyStop)*	patience = 5
		batch_size=1024	monitor=val_accuracy
		val_split=0.2	mode=max

Tabla 6: Variación progresiva de la arquitectura de la Red Neuronal **RNN-LSTM.**

Como se observa en la **Tabla 6**, se resaltan los resultados de cada experimento denotando la mejoría en el desempeño de cada modificación del modelo profundo, siendo el bloque resaltado en verde el mejor desempeño logrado tras las modificaciones de cada experimento. Los resultados obtenidos y los valores de los indicadores de desempeño se analizan en la siguiente sección.

Capítulo 5

Resultados

5.1. Análisis de resultados

5.1.1. Transferencia del Entrenamiento.

Tras encontrar los valores más óptimos para los parámetros de los cuatro modelos de clasificación, se realiza el entrenamiento de los modelos y, mediante las librerías joblib y keras, se guarda en un archivo con el fin de asegurar su consistencia y que el desempeño no varíe con un continuo reentrenamiento. De este modo, aseguramos que el modelo final seleccionado pueda transferirse para su uso en otros entornos.

5.1.2. Comparación de Desempeño

Tras transferir el entrenamiento, se procede a realizar la evaluación de los modelos con la totalidad del corpus, obteniendo los resultados que se observan en la **Tabla 7** a continuación para cada indicador de desempeño.

HYPERPARAMETER SEARCH & MODEL EVAL.		
Classifier	**Best Params.**	**Performance**
Logistic Regression	***C**=1* ***solver**=liblinear* ***penalty**=l1*	**Precision:** 0.8946 **Recall:** 0.9089 **Accuracy:** 0.9010 **F1 Score:** 0.9017
Knearest Neighbors	***n_neighbors**=10* ***weights**=uniform*	**Precision:** 0.7830 **Recall:** 0.8244 **Accuracy:** 0.7981 **F1 Score:** 0.8032
Random Forest	***n_estimators**= 400* ***max_depth**=60*	**Precision:** 0.9414 **Recall:** 0.9513 **Accuracy:** 0.9460 **F1 Score:** 0.9463
Support Vector Machine	***C**=1* ***gamma**=0.1*	**Precision:** 0.9353 **Recall:** 0.9367

	***kernel**=rbf*	**Accuracy:** 0.9207 **F1 Score:** 0.9209

Tabla 7: Evaluación de desempeño de los modelos clásicos.

El modelo de aprendizaje supervisado de mayor desempeño corresponde al Random Forest. Se descartan los modelos restantes y se contrasta el modelo de clasificación seleccionado contra el modelo de aprendizaje profundo refinado. Los resultados obtenidos de este último se reflejan en la tabla a continuación.

HYPERPARAMETER SEARCH & MODEL EVAL.				
Network Version	**Layers**	**Params**	**EarlyStopParms**	**Performance**
BIDIRECTIONAL LSTM 128 Units	Bidirectional LSTM - 128 Units Dense - 64 Units Dense - 1 Unit Dropout (0.5)	epochs=18/20(EarlyStop) batch_size=1024 val_split=0.2	patience = 5 monitor=val_accuracy mode=max	**Testing Accuracy:** 0.9311 **Testing F1 Score:** 0.9169 **Testing Precision:** 0.9228 **Testing Recall:** 0.9509

Tabla 8: Evaluación de desempeño de la Red Neuronal **RNN-LSTM.**

Al observar los resultados de ambos modelos, se observan 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 los modelos para respaldar la selección final.

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

Tabla 9: Costo computacional del entrenamiento de los modelos considerados.

Como se observa en la **Tabla 9**, 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.

Capítulo 6

Desarrollo de la Aplicación

6.1. Requerimientos del Software

Una vez seleccionado el modelo de clasificación más favorable se procede a definir los requerimientos necesarios para el aprovechamiento del modelo dentro del aplicativo.

RF-001	**Cartelera**
Dependencias	-
Descripción	El sistema debe mostrar la lista de las películas presentes en la aplicación.

RF-002	**Indicador de Proporción**
Dependencias	-
Descripción	El sistema debe mostrar una barra que ilustre la proporción de reseñas positivas y negativas de cada película.

RF-003	**Selector de Peliculas**
Dependencias	-
Descripción	El sistema debe contar con un campo selector que permita filtrar la cartelera de películas por los títulos disponibles en la misma.

RF-004	**Botón de Selección de Sección de Reseñas**
Dependencias	RF-001
Descripción	Cada película expuesta en cartelera debe contar con un botón que permita acceder a su sección individual de reseñas.

RF-005	Sección de Reseñas
Dependencias	RF-004
Descripción	Cada película debe contar con una sección individual de sus reseñas

RF-006	Historial de Reseñas
Dependencias	RF-005
Descripción	Cada sección de reseñas debe contar con el histórico de las reseñas previas.

RF-007	Campo de Inserción de Nuevas Reseñas
Dependencias	RF-005
Descripción	Cada sección de reseñas debe contar con un campo editable para insertar nuevas reseñas.

RF-008	Clasificación de sentimiento.
Dependencias	RF-006, RF-005
Descripción	Cada reseña dentro del histórico debe tener una etiqueta que especifique el sentimiento del texto. Cuando se añade una nueva reseña esta debe ser clasificada y añadida al histórico.

RF-009	Retorno a la cartelera
Dependencias	RF – 005
Descripción	La aplicación debe contar con un botón de retorno al menú de películas visible en todas las secciones de la aplicación

6.2. Mockup

Para el diseño de la interfaz gráfica de la aplicación se realizan los mockups de las dos vistas especificadas en los requerimientos. En primera instancia se define la página principal como la cartelera de películas las cuales están desplegadas en una grilla de tarjetas que contiene el título de cada película junto al respectivo botón para acceder a la sección de reseñas correspondiente.

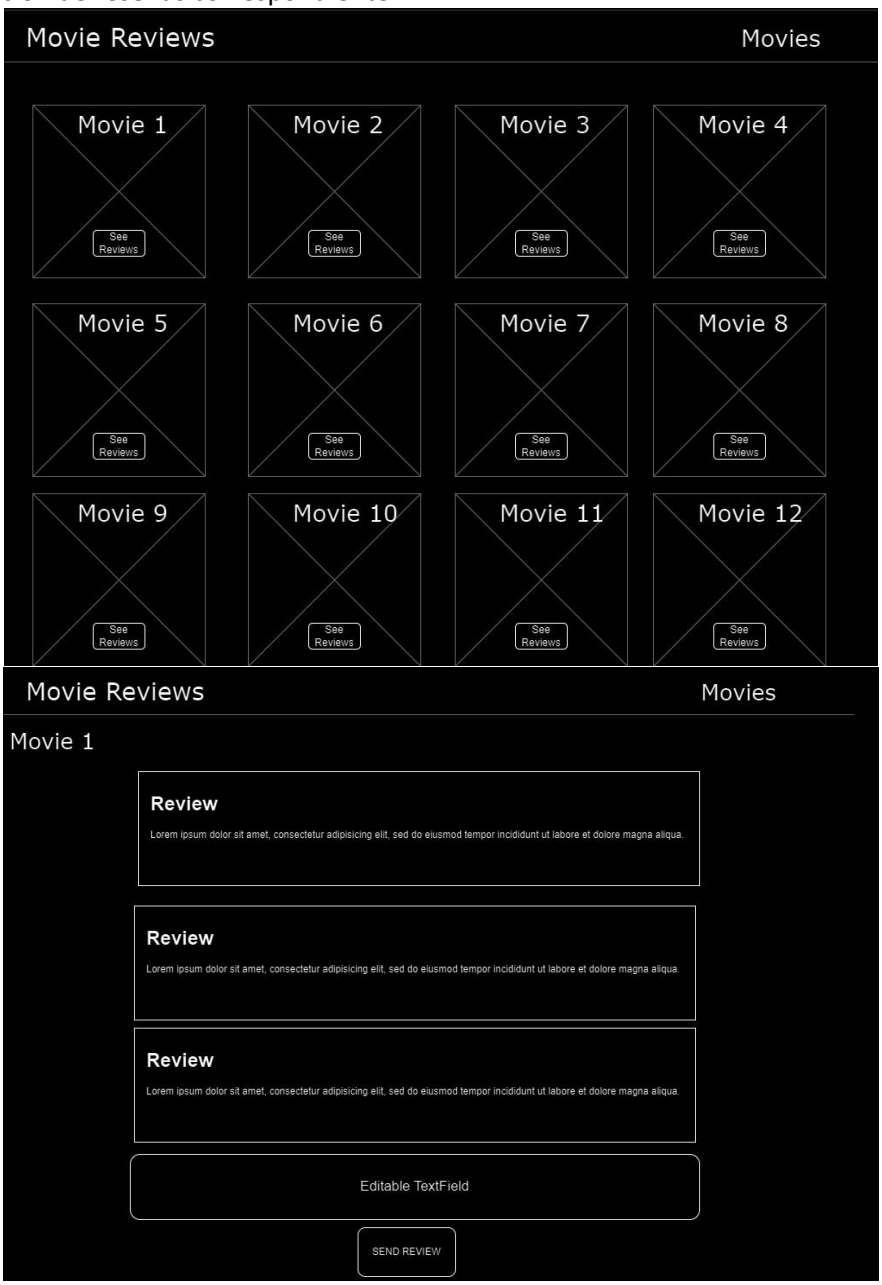


Figura 11: Mockup de la aplicación

Como se observa en la figura, la página principal cuenta con la grilla descrita. Adicionalmente, en la esquina superior derecha se tiene el acceso directo a esta página del aplicativo mediante del botón “Movies” el cual permitirá el retorno a la cartelera en caso de que el usuario entre a alguna sección de reseñas de determinada película.

La vista de la sección de reseñas cuenta con el histórico de las reseñas desplegado verticalmente en donde se aprecia el texto de cada reseña. Al final del histórico de reseñas, está un campo de texto editable que permitirá al usuario ingresar nuevas reseñas. Cuando se añade una nueva reseña esta aparecerá al final del histórico con la clasificación de sentimiento utilizando el modelo seleccionado en la fase experimental.

6.3. Arquitectura de la Aplicación

La arquitectura de la aplicación de la página web de reseñas de películas se divide en tres capas principales: la capa de presentación (frontend), la capa lógica (backend) y la capa de almacenamiento de datos (base de datos). A continuación, se describen los componentes de cada capa:

Capa	Componentes
Capa de Presentación (Frontend)	**Cartelera de Películas:** - Tarjeta con el título de la película. - Botón para acceder a la sección de reseñas de cada película. - Botón en el header para retornar al menú principal/cartelera. **Sección de Reseñas:** - Histórico de reseñas desplegado verticalmente. - Campo de texto editable para añadir nuevas reseñas. - Integración con el modelo de clasificación de sentimiento (Random Forest).
Capa Lógica (Backend)	- Procesar la información relacionada con las películas y reseñas. (título de la película, descripción y texto de las reseñas) - Entrenamiento y clasificación de reseñas en tiempo real. - Integración con la capa de servicios para la clasificación dinámica.
Capa de Almacenamiento de Datos	- Almacenamiento de información sobre películas, reseñas, etc.

Tabla 10: Arquitectura de la Aplicación.

6.4. Implementación del Prototipo

Para la implementación del prototipo se emplearon distintas herramientas y librerías del lenguaje Python. En la tabla se exponen cuales fueron empleadas para el desarrollo de cada capa especificada en la arquitectura de la aplicación:

Capa	Componentes
Capa de Presentación (Frontend)	**Python Dash Plotly:** Dash es un framework para crear aplicaciones web interactivas con Python. Se basa en ***Flask, Plotly y React*** para crear interfaces de usuario interactivas y visualizaciones de datos. Así mismo, permite a los desarrolladores construir aplicaciones web con Python puro, lo que facilita la creación de paneles de control interactivos, cuadros de mando y otras aplicaciones web basadas en datos.
Capa Lógica (Backend)	**Pandas:** Pandas es una biblioteca de Python que proporciona estructuras de datos de alto rendimiento y fáciles de usar, así como herramientas de análisis de datos. **Scikit-learn (sklearn):** Scikit-learn es una biblioteca de aprendizaje automático (machine learning) para Python. Proporciona herramientas simples y eficientes para análisis predictivo de datos y modelado estadístico, incluyendo todos los algoritmos usados en el proceso de experimentación de este proyecto, así como la integración de los mismos al aplicativo.
Capa de Almacenamiento de Datos	**AMAZON S3:** Amazon S3 (Simple Storage Service) es un servicio de almacenamiento en la nube proporcionado por Amazon Web Services (AWS). Un "bucket" en Amazon S3 es un contenedor de objetos (archivos) que se almacenan en S3.

Tabla 11: Tecnologías de Implementación.

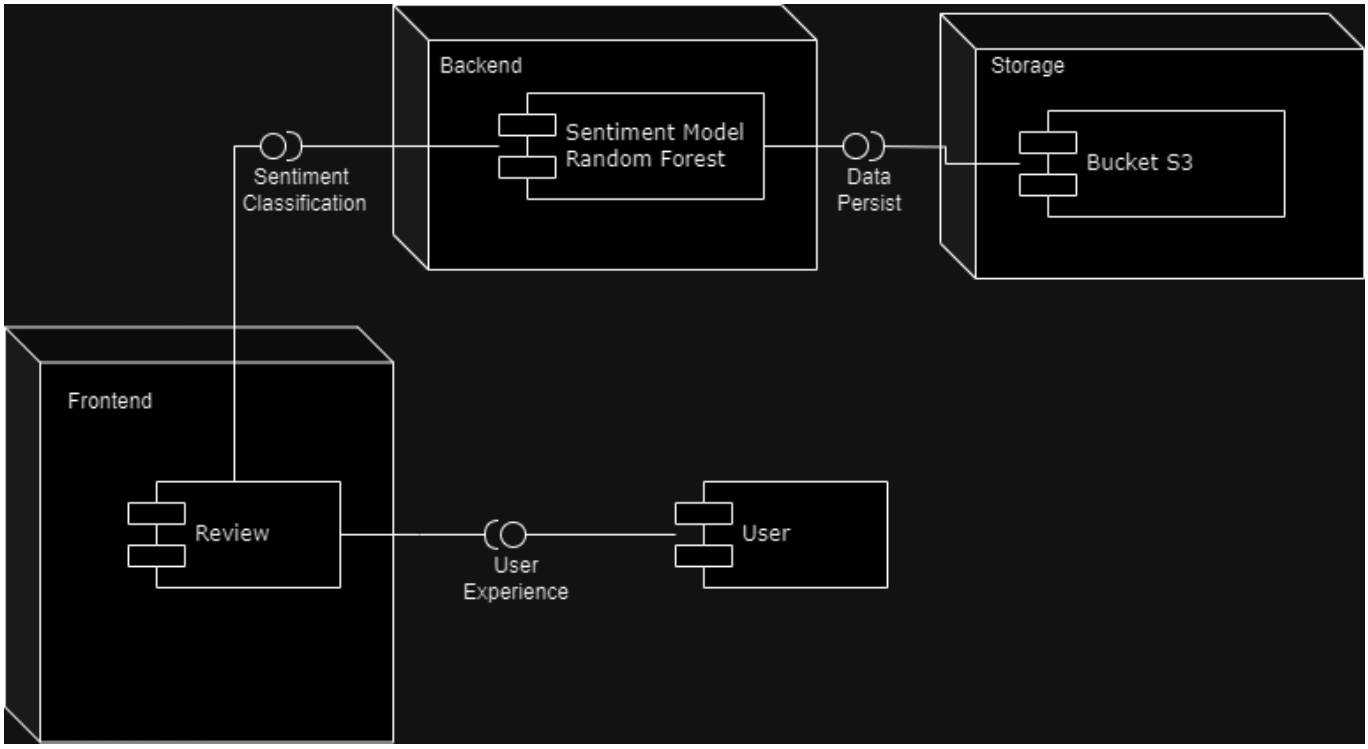


Figura 11: Diagrama de Componentes

Como se expuso en los requerimientos **RF-001**, **RF-002**, **RF-003** y **RF004**; la página principal del aplicativo debe contar con una grilla de películas, un selector para filtrar por título, un indicador de proporción de las reseñas y un botón disponible en cada tarjeta de película para acceder a su sección individual de reseñas. En la siguiente figura se observa la implementación de la vista de la cartelera de películas.

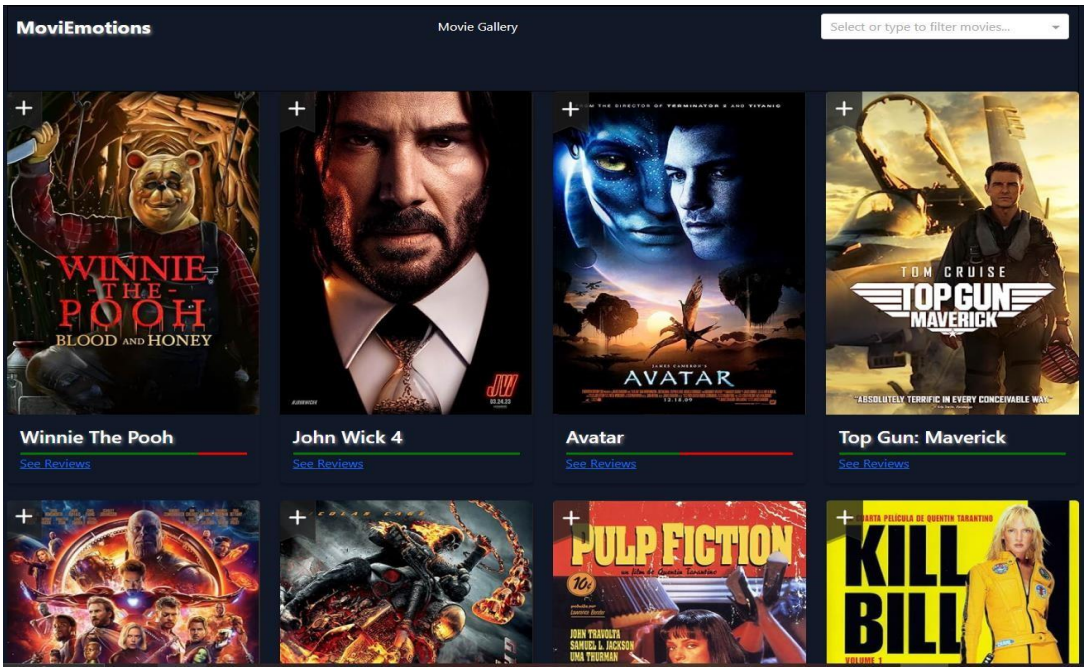


Figura 12: Vista - Cartelera de Películas

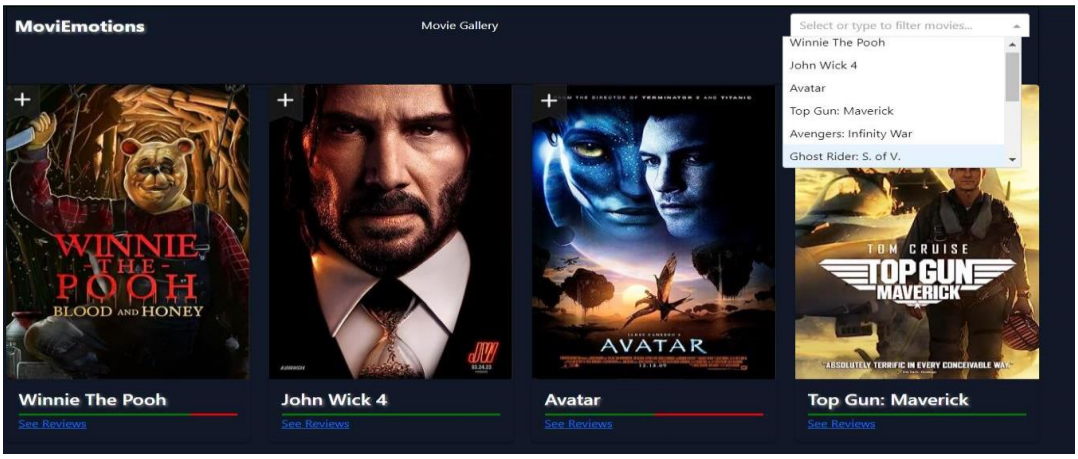


Figura 13: Selector de Películas - Desplegable

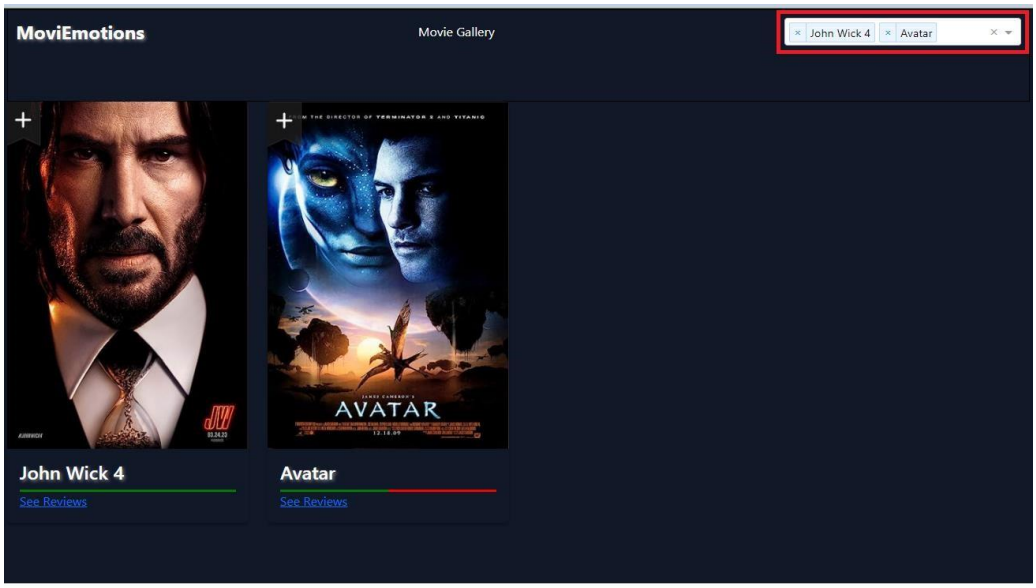


Figura 14: Selector de Películas - Filtro

Continuando con la implementación de las vistas, tal como se expone en los requerimientos **RF-005**, **RF006**, **RF-007**, **RF-008 y RF009** cada película cuenta con su sección individual de reseñas compuesto por el histórico de reseñas en donde cada texto va acompañado de su etiqueta de sentimiento. Así mismo, se cuenta con un campo para ingresar nuevas reseñas y un botón de retorno a la sección de cartelera ("Movie Gallery"). En la siguiente figura se observa la implementación de la vista de la sección de reseñas.

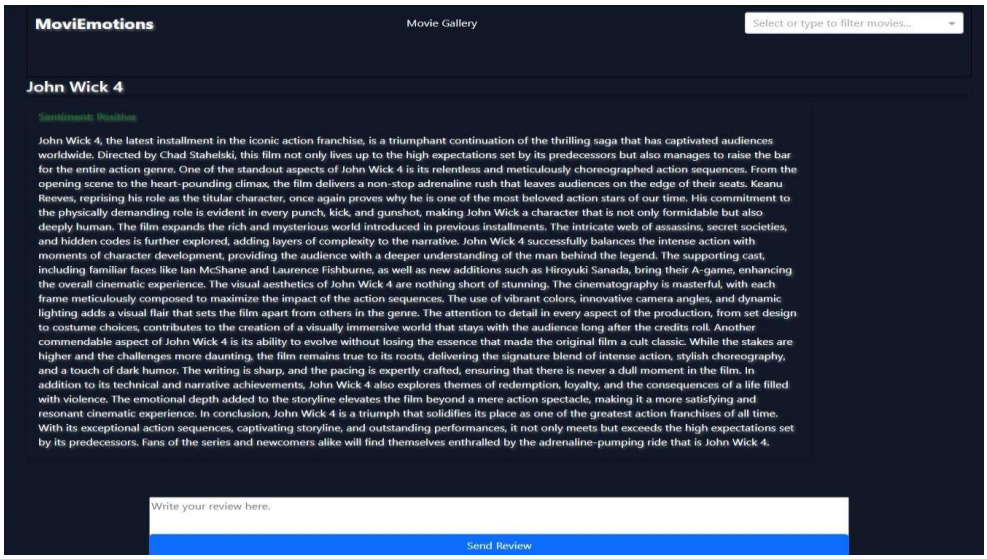


Figura 15: Vista - Sección de Reseñas

El campo para escribir nuevas reseñas añade el texto al final del histórico y clasifica el mismo de manera dinámica adjuntándole la etiqueta correspondiente al sentimiento obtenido por el modelo integrado en el aplicativo.

6.5. Casos de Prueba

Se realizan las pruebas del sistema tras implementar el primer prototipo usable. A continuación, se describen los casos de prueba ejecutados para los dos módulos de la aplicación.

CASO DE PRUEBA			
Caso de Uso:	Filtro de películas	**ID:**	P001
Descripción:	Busqueda de películas por medio del selector de títulos de los filmes.	**Módulo:**	Cartelera
CASO DE PRUEBA - P001			
Precondiciones:	Deben existir peliculas en la cartelera, el selector debe proporcionar los títulos existintes de los filmes.		
Datos de Entrada:	-		
Secuencia:	Desplegar el selector, Seleccionar el título de interés, adjuntarlo al filtro de a cartelera		
Postcondiciones:	La grilla/cartelera de películas debe ser actualizada siendo solo visibles los títulos filtrados.		
RESULTADOS			
Observaciones	-		
Resultado	Con Éxito.		

CASO DE PRUEBA			
Caso de Uso:	Selección de Sección de Reseñas	**ID:**	P002
Descripción:	Selección de sección individual de reseñas de un filme.	**Módulo:**	Cartelera
CASO DE PRUEBA - P002			
Precondiciones:	Deben existir peliculas en la cartelera, cada tarjeta de pelicula debe contar con su botón de reseñas.		
Datos de Entrada:	-		
Secuencia:	Seleccionar el botón de sección de reseñas de la tarjeta de alguna película		
Postcondiciones:	Se debe producir una transición desde la cartelera a la sección de reseñas seleccionada.		
RESULTADOS			
Observaciones	-		
Resultado	Con Éxito.		

CASO DE PRUEBA			
Caso de Uso:	Inserción de Nueva Reseña	**ID:**	P003
Descripción:	El usuario inserta el texto de una nueva reseña y lo envía al sistema.	**Módulo:**	Sec. de Reseñas
CASO DE PRUEBA - P003			
Precondiciones:	Debe existir un campo de inserción de texto y un botón de envío que permita enviarlo al sistema.		
Datos de Entrada:	Texto proporcionado por el usuario		
Secuencia:	El usuario inserta el texto de una nueva reseña, da click al botón de envío para subirlo al sistema.		
Postcondiciones:	El texto de la nueva reseña se adjunta al histórico de las reseñas.		
RESULTADOS			
Observaciones	-		
Resultado	Con Éxito.		

CASO DE PRUEBA			
Caso de Uso:	Clasificación de Sentimiento	**ID:**	P004
Descripción:	El sistema clasifica una reseña enviada por un usuario.	**Módulo:**	Sec. de Reseñas
CASO DE PRUEBA - P004			
Precondiciones:	El sistema debe haber recibido una nueva reseña proporcionada por un usuario.		
Datos de Entrada:	Texto proporcionado por el usuario		
Secuencia:	El usuario envía un nuevo texto al sistema, el texto es procesado y transformado en vector numerico para su posterior clasificación haciendo uso del modelo Random Forest.		
Postcondiciones:	El texto del usuario es adjuntado al histórico de reseñas con su respectiva etiqueta de sentimiento. El indicador de proporción de reseñas es actualizado manteniendo la relación del color verde indicando la proporción de reseñas positivas y el rojo para las negativas.		
RESULTADOS			
Observaciones	Se realizan tres iteraciones del caso. En primera instancia se añaden incidencias de la clase positiva unicamente, seguidamente se tratan incidencias de la clase negativa unicamente y se finaliza con una incidencia de cada clase con el fin de evaluar el funcionamiento del indicador de proporción.		
Resultado	Con Éxito.		

CASO DE PRUEBA			
Caso de Uso:	Retorno a la Cartelera	**ID:**	P005
Descripción:	Selección del botón de retorno a la selección de películas	**Módulo:**	Sec. de Reseñas
CASO DE PRUEBA - P005			
Precondiciones:	Se debe encontrar en la sección de reseñas de alguna película, Debe existir un botón que permita retornar desde una sección de reseñas a la cartelera		
Datos de Entrada:	-		
Secuencia:	Dar click al botón de retorno a la cartelera.		
Postcondiciones:	Se debe producir una transición desde la sección de reseñas a la cartelera.		
RESULTADOS			
Observaciones	-		
Resultado	Con Éxito.		

Capítulo 7

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 vectorizador.

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] Alpaydın, E. (2010). *Introduction to Machine Learning: Adaptive Computation and Machine Learning*. En *What is Machine Learning?* (2.a ed., Vol. 1). The MIT Press Cambridge, Massachusetts London, England.

[3] Simeone, O. (2018). A Very Brief Introduction to Machine Learning With Applications to Communication Systems. IEEE. https://arxiv.org/pdf/1808.02342.pdf

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

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

[6] 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

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

[8] Random Forest Lakshmanaprabu, S.K., Shankar, K., Ilayaraja, M. et al. Random forest for big data classification in the internet of things using optimal features. Int. J. Mach. Learn. & Cyber. 10, 2609–2618 (2019). https://doi.org/10.1007/s13042-018-00916-z

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

[10] Qin, T. (2021, 14 noviembre). Dual Learning (2020 ed.). Springer**.**

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

[12] Beysolow, T. (2018). *Applied Natural Language Processing with Python: Implementing Machine Learning and Deep Learning Algorithms for Natural Language Processing*. En *What is Natural Language Processing?* (1.a ed., Vol. 1). Apress.

[13] Oxford Languages and Google - Spanish | Oxford Languages. (2022, 15 febrero). Recuperado 4 de octubre de 2022, de https://languages.oup.com/google-dictionary-es/

[14] López Plazas, N. C. (s. f.). Reseña de una Película. Universidad de los Andes. https://leo.uniandes.edu.co/images/Guias/Resea-pelcula.pdf

[15] Soubraylu, S. & Rajalakshmi, R. (2019). Hybrid convolutional bidirectional recurrent neural network based sentiment analysis on movie reviews. School of Computer Science andEngineering, Vellore Institute ofTechnology, Chennai, India.
https://onlinelibrary.wiley.com/doi/epdf/10.1111/coin.12400

[16] Qaisar, S. M. (2020). Sentiment Analysis of IMDb Movie Reviews Using Long Short-Term Memory. Qaisar. College of Engineering, Effat University.
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9257657

[17] Novelty Octaviani, A. (2020). Sentiment Analysis on Movie Reviews Using Information Gain and K-Nearest Neighbor. Telkom UniversityBandung, Indonesia.
http://commdis.telkomuniversity.ac.id/jdsa/index.php/jdsa/article/view/22/18

[18] Dashtipour, K., Gogate, M., Adeel, A., Larijani, H. & Hussain, A. (2021). Sentiment Analysis of Persian Movie Reviews Using Deep Learning. , University of Stirling.
https://www.mdpi.com/1099-4300/23/5/596/pdf