Sistema basado en procesamiento de textos para la predicción de la aceptación de películas a partir de su guión

1st Aco Guerrero Iván Rogelio
Facultad de Ingeniería
UNAM
México, CDMX
ivan-rogelio@hotmail.es

2nd García Fernández Jesús Alejandro
Facultad de Ingeniería
UNAM
México,CDMX
ale.garcia.fernandez@hotmail.com

3nd Hernández Arrieta Carlos Alberto
Facultad de Ingeniería
UNAM
México,CDMX
carlos.arrieta.merec@gmail.com

4th Ramírez Castillo Miguel Ángel
Facultad de Ingeniería
UNAM
México,CDMX
migram.cas5@gmail.com

***Abstract*—This document shows the result of a system developed to predict the rating of a movie receiving only subtitles as input. Bayesian classifier, Laplace classifier, SVM classifier and K-Neighbors classifier were used.**

***Index Terms*—clasificador, películas, bayes, svm, laplace, K-neighbors**

I. Introducción

A. Motivación

Las artes son de las actividades más importantes para la vida del ser humano, pues mientras las áreas de ingeniería se enfocan a facilitar herramientas a las personas con el fin de mejorar su calidad de vida, las artes aspiran a darle un sentido a ésta.

Y efectivamente, sin la arquitectura, escultura, pintura, música, literatura, danza y, sobre todo para este caso en particular, el cine, los seres humanos vivirían sumergidos en la vida sin sentido alguno. Además, las bellas artes les han permitido construirse y deconstruirse para experimentar como seres vivos la búsqueda de ellos mismos y ha posibilitado un canal más para que ante el mundo exprese sus sentimientos.

Al analizar esta situación, de manera conjunta se logró percibir que un filme, aunque puede llegar a emocionar a muchas personas en su estreno, también nos puede permitir construir un artefacto que permita identificar qué películas llegarán a ser mejor aceptadas por las personas por medio de una calificación, con respecto a otras antes de que sean estrenadas, por lo que la curiosidad y la ambición de predecir situaciones futuras en el mundo cinematográfico ha sido la motivación par entrelazar los conocimientos de ingeniería y cine y hacer realidad esta interesante tarea.

B. Justificación

El presente trabajo aspira a demostrar que es posible estimar la calificación de una película conociendo únicamente su guión, para ello esta actividad se cimenta sobre una sólida investigación que el lector podrá descubrir a lo largo de sus hojas, en donde cada una de ellas refuerza el argumento inicial: predecir si tendrá éxito el filme. Este trabajo será útil para identificar qué tanto peso tiene el guión de una película con respecto a los demás aspectos que puede ofrecer una obra de este estilo. Si la película coincide con la calificación dada en sitios como IMDB podemos a llegar a suponer que el guión tiene un gran peso en la obra. Por otro lado, se pretende detonar la exploración y la posterior invención de instrumentos que puedan predecir el éxito que tendrá una idea al materializarse o simplemente para analizar la posibilidad de que ocurran ciertos eventos a futuro, ya sea para fines educativos, de investigación o comerciales. Como equipo no solo vemos la utilidad de este trabajo en el ámbito cinematográfico sino que cuenta con el potencial suficiente para verterse en cualquier otra área donde pueda ser explotado, por lo que este documento es por sí mismo una demostración de que es posible la construcción de tales herramientas.

C. Problema

El mundo cinematográfico constantemente crea filmes que pueden llegar a tener una excelente valoración por parte de los espectadores o ser duramente criticada, por lo que se pretende a partir del guión de una película, predecir el nivel de aceptación que tendrá ésta entre la audiencia, teniendo como referencia el nivel de aceptación de otras ya calificadas.

D. Solución Propuesta

Se propone el desarrollo de un sistema que permita al usuario ingresar un documento de texto con los subtítulos o guión de una película, de la cual se desee conocer su calificación.

Esta herramienta primero realizará el procesamiento de un conjunto de guiones que permitirán entrenar al sistema. Para ello se inicia por la limpieza de cada uno de éstos (se elimina todo carácter que no sea una letra); posteriormente se les aplicará stemming para reducir una palabra a su raíz, y luego se procederá a obtener las frecuencias de las palabras y se aplicarán las fórmulas correspondientes para generar la tabla TF, IDF y por último la TF-IDF que hace uso de las dos anteriores. Cabe destacar que los conjuntos de guiones formarán cuatro grupos diferentes: aceptación mala, regular, buena y excelente; y toda la información estará contenida en una sola matriz TF-IDF. Ahora bien, dicha tabla se usará para entrenar al algoritmo que realizará las predicciones.

Por otra parte, el conjunto de textos que se mencionaron recientemente se definirán con base en la calificación del sitio IMDB. Ahora bien, los documentos como tal serán adquiridos a través del sitio web Opensubtitles.com, ya que es una página que tiene una alta cantidad de guiones en varios idiomas con una descarga totalmente gratuita. Bien vale la pena mencionar que se pretenden trabajar solamente con archivos en inglés.

Ya que se tiene entrenado el sistema predictor, se pretende que el usuario introduzca el guión de una película, preferentemente descargada de OpenSubtitles.com o un sitio similar, y el software hará un proceso muy similar al que aplica con los archivos de entrenamiento, es decir, limpieza del documento, aplicación de stemming, obtención de frecuencias de cada palabra, etc. Y con algún algoritmo de clasificación, entre los que destacan SVM, Bayesiano o N-vecinos ya se obtiene el resultado. Al final, el sistema simplemente dará una calificación o nivel de aceptación que se estima que tendrá la película.

Al obtener el resultado, se pondrá en duda si la calificación arrojada es cercana a la calificación que obtendrá la película por parte de los críticos. Es importante mencionar que podrá no ser la misma, porque la crítica que recibe una película va mucho más allá de lo que está escrito en un guión, pero resultará un ejercicio bastante interesante intentar estudiar qué tan próximo puede ser una predicción contra lo que se maneja en una evaluación completa

II. Marco Teórico-Metodológico

A. Antecendentes

La valoración de una película es determinada por una calificación dada por un conjunto de críticos y la audiencia que haya visto la película. Un caso es *Rotten Tomatoes* [1], página web que elabora reseñas de películas, aquí la valoración de una película es el porcentaje de aceptación de un conjunto de opiniones de críticos, por ejemplo una película puede tener 70% de aceptación. Para *IMDB* [2], la base de datos de películas más grande del mundo, la valoración de una Película es obtenida a partir del promedio de Ratings dado por los usuarios, el rating que puede dar cada usuario es del 1 al 10, por ejemplo una película puede tener un Rating de 7.8. Dicho lo anterior, se puede observar que ambos sistemas de valoración están basados en la opinión de la gente, qué tanto les gustó una película, esta es la gran diferencia entre el sistema a desarrollar en este texto y los sistemas de puntuación tradicionales, ya que se desea predecir la valoración de una película sin que sea necesario la opinión de las personas,sino únicamente basándose en el guión de una película.

Como se mencionó en la sección de Solución Propuesta, para llevar a cabo el sistema de valoración de una película vamos a tomar como base dos herramientas de Análisis de Textos:

- TF-IDF
- Clasificador Bayesiano

B. TF-IDF (term frequency-inverse document frequency)

Es una medida estadística que permite evaluar qué tan relevante es una palabra de un documento dentro de un conjunto de documentos [3]. Esta técnica es ampliamente usada mucho en el Procesamiento Natural de Lenguaje debido a que permite asignar a cada palabra un determinado puntaje dado que tan relevante es la palabra, para posteriormente usar este como entrada para otros algoritmos.

Esta técnica se constituye de dos partes:

- *Term Frecuency*: Es la frecuencia de una palabra en un documento.
- *Inverse Document Frecuency*: Esta medida nos permite saber qué tan inusual es una palabra dentro del conjunto de documentos, si la palabra tiene un puntaje cercano a 0, entonces es una palabra muy común, de lo contrario es una palabra muy inusual.

Juntando ambas técnicas se calcula el puntaje *TF-IDF* de una palabra w del documento d dentro del conjunto de documentos D de la siguiente forma:

$$TF - IDF(w, d, D) = tf(w, d) * idf(t, D) \qquad (1)$$

Si una palabra aparece mucho en un documento (TF alto) y poco en otros documentos (IDF alto) entonces la palabra es relevante.

Esta herramienta es de alta relevancia dentro del problema a atacar porque permite que las palabras importantes tengan más peso al momento de realizar la clasificación de la Película.

C. Bag-of-Words Model

Es un método que permite extraer las características de un texto para ser usado en un modelo, en este caso para el *Naïve Bayes Classifier*. Formalmente es una representación que describe la ocurrencia de las palabras dentro del documento,

dónde la información del orden y estructura de las palabras es descartado [5].

El siguiente ejemplo describe cómo funciona este modelo:

1) Tenemos los siguientes documentos:
   d_1 = "Hola, cómo estás"
   d_2 = "Hola, qué tal"
2) El primer paso es diseñar el vocabulario de palabras conocidas: V = {Hola, cómo, estás, qué, tal}
3) El segundo paso es inicializar el vector v_n de cada documento d_n:
   Para d_1, el vector es v_1 = {Hola: 0, cómo: 0, estás: 0, qué: 0, tal: 0}
4) El tercer paso es contar el número de ocurrencias de cada palabra de un vector v_n en el documento d_n.
5) Para el vector v_1 con el documento d_1:
   v_1 = {Hola: 1, cómo: 1, estás: 1, qué: 0, tal: 0}
6) Para el vector v_2 con el documento d_2:
   v_2 = {Hola: 1, cómo: 0, estás: 0, qué: 1, tal: 1}
7) Al conjunto de vectores $\{v_1, v_2\}$ se les conoce como la bolsa de palabras.

D. Algoritmos

1) Naïve Bayes Classifier: Es un clasificador probabilístico usado para obtener la clase c a la que pertenece un objeto i [4].

Para determinar a qué clase c pertenece un documento d, obtenemos la probabilidad de que dado un documento d, pertenezca a la clase c:

$$p(c|d) = \frac{p(d|c)p(c)}{p(d)} \tag{2}$$

La clase c que sea más probable para el documento d es aquella que tenga el valor de probabilidad más grande.

2) Multinomial Naïve Bayes Classifier: Es una variante del clasificador anterior dónde se tiene más de una variable de entrada y estas variables son independientes.

Usando este modelo, ya no sólo se depende de una variable de entrada sino de varias variables de entrada, por lo que se puede describir la probabilidad de d dado c como la probabilidad de un conjunto de características x dado c :

$$P(d|c) = P(x_1, x_2, ..., x_n|c) = P(x_1|c)P(x_2|c)...P(x_n|c) \tag{3}$$

Utilizando el *Bag-of-Words Model*, cada x de la fórmula anterior es la ocurrencia de cada palabra dentro del documento d.

Para calcular la probabilidad de $P(c|d)$ es a través de las siguientes fórmulas:

La probabilidad de la clase c es:

$$\hat{P}(c) = \frac{N_c}{N} \tag{4}$$

Dónde N_c es el número de documentos de la clase c y N es el número de documento total.

La probabilidad de un palabra w dada la clase c es:

$$\hat{P}(w|c) = \frac{count(w,c)+1}{count(c)+|V|} \tag{5}$$

Dónde $count(w,c)$ es la ocurrencia de la palabra w dentro de los documentos con clase c, $count(c)$ es el número de palabras de los documentos con clase c y $|V|$ es el tamaño del vocabulario.

Finalmente la probabilidad de la clase c dado el documento d es proporcional a lo siguiente:

$$P(c|d) \propto \hat{P}(w_1|c) * \hat{P}(w_2|c) * ... * \hat{P}(w_n|c) \tag{6}$$

Para la clase c_n, dónde $P(c_n|d)$ sea el valor más alto, es la clase que le vamos a asignar al documento d.

E. Clasificador K vecinos

El clasificador K vecinos es un algoritmo usado para realizar la clasificación de conjuntos. El algoritmo *KKN(k-nearest neighbors algorithm*) supone que existen cosas similares en las proximidades. En otras palabras, cosas similares significa que están cerca una de la otra. *KNN* puede usarse tanto para problemas predictivos de clasificación como de regresión. Sin embargo, se usa más ampliamente en la industria en problemas de clasificación [6].

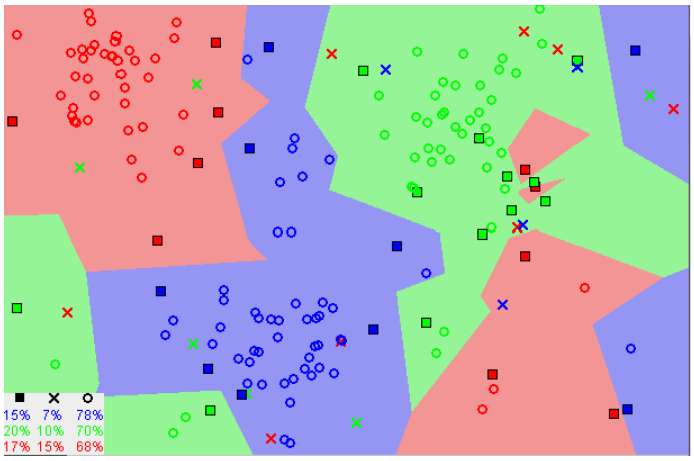

Fig. 1. Ejemplo de *KNN*

La idea es sencilla, el algoritmo clasifica cada dato nuevo en el grupo al que pertenece, según tenga k vecinos más cerca de un grupo o de otro. Es decir, calcula la distancia del elemento nuevo a cada uno de los existentes, y ordena dichas distancias de menor a mayor para ir seleccionando el grupo al que pertenece. Este grupo será, por tanto, el de mayor frecuencia con menores distancias.

El *KNN* es un algoritmo de aprendizaje supervisado, es decir, que a partir de un juego de datos inicial su objetivo será

el de clasificar correctamente todas las instancias nuevas. El juego de datos típico de este tipo de algoritmos está formado por varios atributos descriptivos y un solo atributo objetivo, también llamado clase. Los pasos para desarrollar el algoritmo son los siguientes:

- Cargar los datos
- Inicializar K en un número de vecinos elegido
- Para cada dato en el conjunto de datos
  - Calcule la distancia entre el dato de consulta y el dato actual de los datos.
  - Agregue la distancia y el índice del dato a una colección ordenada.
- Ordene la colección ordenada de distancias e índices de menor a mayor (en orden ascendente) por las distancias
- Elija las primeras K entradas de la colección ordenada
- Obtenga las etiquetas de las entradas K seleccionadas
- Devuelva las etiquetas K

F. Clasificador SVM

El objetivo del algoritmo *SVM* es crear la mejor línea o límite de decisión que pueda segregar el espacio n-dimensional en clases para que podamos colocar fácilmente el nuevo punto de datos en la categoría correcta en el futuro. Este límite de mejor decisión se llama hiperplano. Esto se refiere a que el conjunto de datos es representados en un espacio 2D para luego ser separado por una línea que dé como resultado los conjuntos separados.

SVM elige los puntos/vectores extremos que ayudan a crear el hiperplano. Estos casos extremos se denominan vectores de soporte y, por lo tanto, el algoritmo se denomina Máquina de vectores de soporte.

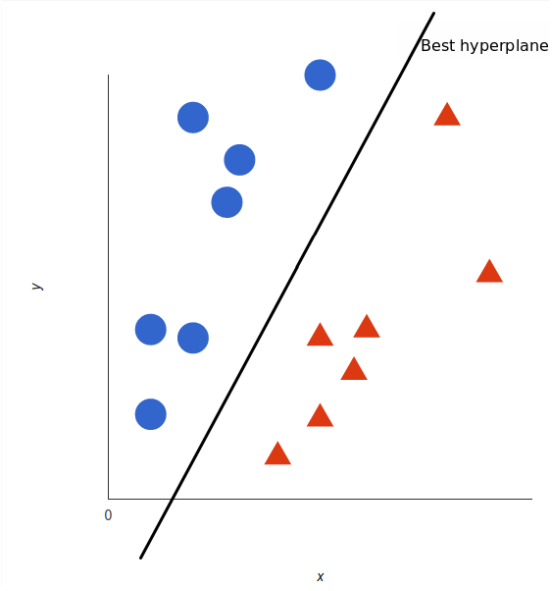

Fig. 2. Ejemplo de *Clasificador SVM de dos dimensiones*

Pero hay casos que no son tan sencillos como el mostrado arriba donde claramente había una opción donde la línea podía separar los dos conjuntos, en este caso nace una. En esos otros casos podemos auxiliarnos por más dimensiones. El algoritmo SVM buscará la forma de crear una línea que separe los conjuntos.

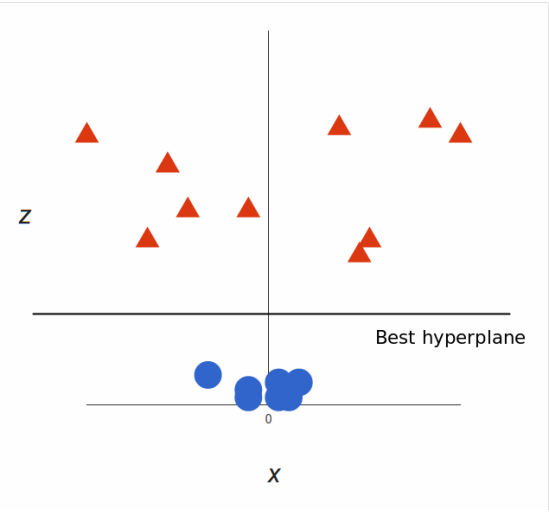

Fig. 3. Ejemplo de *Clasificador SVM de tres dimensiones*

¿Cómo se realiza una predicción actualmente?

Es importante señalar que actualmente no existe ningún sistema que se asimile al funcionamiento del sistema propuesto; hoy en día, todas las predicciones realizadas en la industria del cine se realizan a partir del conocimiento y estudio de personal experto en el área, en donde, sin lugar a dudas, se presenta en algún punto la opinión personal del experto aunque sea en el menor nivel.

Por lo tanto, se desarrollará la propuesta mencionada tomando como base sólo los conocimientos personales y, evaluando la confiabilidad del sistema propuesto, sometiendo al análisis guiones de películas ya calificadas y comparando el resultado del sistema con la calificación asignada en la página IMDB.

G. Trabajos relacionados

Durante los últimos años se han desarrollado una gran cantidad de sistemas capaces de realizar predicciones o descripciones a partir de un conjunto de datos o textos. Este trabajo se centró principalmente en construir un modelo predictivo que sea capaz de determinar la probabilidad de que una película, previo a que salga al público, sea catalogada como mala, regular, buena o excelente con base en su diálogos.

Previamente a la materialización de esta idea, ya existían sistemas similares, y aunque no fue posible encontrar un trabajo igual al que se describe en este documento, bien es cierto que se han construido dichos sistemas en otras áreas, por lo que a continuación se documentan varios de estos casos. Todos ellos tienen la característica que permitieron ayudar a entender y aclarar las dudas sobre el procedimiento que se debe seguir para formular un modelo de esta índole.

1) Una aproximación conociendo datos precisos de la película: Existen diversas bibliotecas que han abordado este problema. Por ejemplo, en Kaggle.com existe un predictor de score de películas que deduce la calificación a partir de los siguientes parámetros:

Su Método hace uso del algoritmo de random forest. Es bastante diferente al método presentado en este escrito ya que el utiliza parámetros específicos de la película para conocer qué calificación puede tener en lugar de sólo conocer sus subtítulos. Ambas soluciones hacen uso de un sistema de entrenamiento.

2) Sistema de búsqueda personalizada y recomendación de documentación científica: El proyecto estuvo a cargo de Erika J. Salazar y Oscar Ortega L., para la universidad de Antioquia en Medellín, Colombia.

Básicamente se desarrolló un sistema de búsqueda y recomendación automática de documentos, dirigido hacia los usuarios investigadores de una comunidad académica con intereses de información documental especializada. El sistema cuenta con varios módulos, por una parte está el módulo de generación de consultas para extraer y transformar en consultas los términos más importantes que se encuentren en el perfil del usuario que haya hecho dicha búsqueda; el módulo de búsqueda y descarga de documentos para que se envíen las consultas a distintos buscadores de documentos científicos y descargarlos; otro módulo es el de agrupamiento, cuyo objetivo es el de procesar y almacenar los documentos obtenidos, y por último, un módulo de filtrado, que como su nombre lo indica, filtra los subconjuntos de documentos relevantes para ser recomendados a otros usuarios y ajustar los perfiles a partir de las calificaciones que los usuarios suministran a los documentos que le son recomendados.

Para poder realizarlo se consideraron varios métodos, entre ellos están Boley, TF-IDF y TF. J. Salazar (2006) explica que en el sistema, los perfiles de usuario y los documentos son representados como vectores de frecuencias de términos TF, y se reducen a sus raíces utilizando stemming y eliminación por stop-words, sin embargo no se obtiene el IDF porque al ser un sistema con flujo dinámico de documentos, calcularlo costaría el reprocesamiento de las frecuencias de todos los términos del sistema para todos los documentos nuevos que se descarguen, por que lo que se optó simplemente por TF para la generación de consultas.

3) Una propuesta para la clasificación emocional de un álbum a partir de la letra de sus canciones: La presente investigación fue desarrollada por Francisco Javier Moreno Arboleda, John Freddy Duitama Muñoz y Luis Fernando Montoya Gómez para la Universidad Distrital "Francisco José de Caldas".

Actualmente hay muchos sitios en internet que ofrecen servicios de música como www.musicovery.com o frendlymusic.com. Todos ellos tienen en común el uso de categorías emocionales, pues les permite a las plataformas crear listas de reproducción y encontrar canciones relacionadas al estado de ánimo de los usuarios. Para lograr el objetivo se pueden llegar por dos caminos, la primera alternativa se refiere a la retroalimentación de los usuarios y una segunda opción es obtener esta clasificación a partir de características inherentes que tienen las canciones tales como la letra y sonido.

Aunque para esta última opción es necesario aplicar técnicas de audio, para analizar la letra se requieren técnicas de minería de textos. Sin embargo, cabe destacar que según sus investigaciones, previamente otros autores demostraron que la incorporación de características espectrales del audio no mejora de forma significativa la precisión de la clasificación, por esta razón se concentraron más en el texto.

Los pasos que siguieron de forma muy general fueron los siguientes:

- Preprocesamiento de las canciones: Se convierte el texto a minúsculas, se eliminan saltos de línea y todos los caracteres que no sean letras.
- Eliminan las stopwords: Se quitan todas las palabras que aportan poco o nada de significado para clasificar
- Aplicar stemming: De esta forma se obtiene la raíz de la palabra, para ello se utilizó el algoritmo de Snowball.
- Uso de los SVM (Support-vector machines): son algoritmos de aprendizaje supervisado, que sirven para tareas de clasificación y regresión basados en datos de entrenamiento, sin embargo, como funcionan con parámetros numéricos de entrada y salida, se debe convertir el texto en un formato válido.
- Uso de TF-IDF: Es una medida que expresa qué tan relevante es un término para un documento con relación a un conjunto de documentos; y fue de utilidad para poder convertir las letras de las canciones en parámetros válidos para los algoritmos SVM.
- Para clasificar se crearon 18 categorías emocionales , y se seleccionaron 30 canciones para cada categoría.

4) La recuperación de información en español y la normalización de términos: El trabajo de investigación dirigido por Carlos G. Figuerola, Ángel F. Zazo, Emilio Rodríguez Vázquez y José Luis Alonso Berrocal para la universidad de Salamanca, España, como tal no buscaba construir un sistema que realizara alguna predicción a partir de un conjunto de textos, más bien fue el objetivo era demostrar la importancia de la normalización de los términos antes de utilizar esos textos en los algoritmos y evaluar su eficacia.

Pues bien, concluyeron que, por lo menos en el idioma español, la normalización de términos mejora considerablemente los resultados en la recuperación de información, sobre todo cuando las consultas son cortas. Además, para poder realizar una comparación usaron n-gramas, pero determinaron que resultan desaconsejables pues los resultados que se obtienen no alcanzan a los que se registran aún sin aplicar ningún tipo de normalización. Y por último, los algoritmos más complejos no consiguen superar los resultados obtenidos por un simple s-stemmer, que es más fácil de implementar.

III. Método experimental

A. Datos de entrenamiento y prueba

Con anterioridad se mencionó ligeramente el tema sobre el origen de los datos que se van a implantar en el sistema tanto para entrenamiento como para pruebas. Básicamente provienen de dos sitios web: opensubtitles.com e IMDB.

La página opensubtitles.org es un sitio web que contiene millones de subtítulos de películas en diferentes idiomas, diálogos que fueron escritos o traducidos por usuarios de internet con la finalidad de compartirlos al público para que puedan integrarlos a sus películas rompiendo así la barrera del idioma para cualquier usuario de internet.

Opensubtitles tiene muchas bondades para nosotros, más allá de que recopila los archivos creados por otros usuarios o subidos directamente por ellos, también cuenta con búsquedas avanzadas para filtrar más rápidamente un conjunto de películas por idioma, género, año y sobre todo, puntuación. Sobre este último punto vale la pena mencionar que a pesar de que la plataforma cuente con una valoración proveniente supuestamente de la plataforma IMDB, hay casos en que las calificaciones no concuerdan ya que posiblemente no se han actualizado, por lo que para subsanar este error y así tener datos más fidedignos, optamos por consultar directamente esta plataforma para obtener las evaluaciones y la otra únicamente para descargas.

Una primera dificultad se presenta al homologar la escala del sitio IMDB con la nuestra, ya que mientras nosotros manejamos 4 categorías, como se mencionó previamente (malas, regulares, buenas y malas), la plataforma trabaja con una escala numérica que se encuentra en un rango de 0 a 10 estrellas. A consecuencia de ello se propusieron las siguientes equivalencias:

Rango de Calificación IDMB	Clasificación del sistema de predicción
0 - 5.0	Mala
5.0-6.67	Regular
6.67 a 8.34	Buena
8.34-10	Excelente

En la tabla anterior se optó por considerar como malas a películas dentro de un rango de 0 a 5 de calificación en la escala de IMDB, y esto se debe principalmente a una cuestión de cualidad, ya que se llegó a la conclusión que no puede ser considerada como regular una película con una calificación inferior a 5, pues cuando se piensa en algo con esta calificación, como por ejemplo, las evaluaciones en la escuela, generalmente sobrepasa este valor. Básicamente fue la razón por la que se sesgó la escala.

Para entrenar el modelo se descargaron 100 guiones de películas por cada clasificación a través de la plataforma ya mencionada, por lo que al ser cuatro categorías se descargaron 400 subtítulos en total. Ahora bien, para realizar pruebas se descargaron otras 8 películas distintas a las 400 descargadas anteriormente; correspondientes a dos por cada categoría de la escala.

Cabe mencionar que se tomaron los filmes que aparecen en los Top de la plataforma IDMB sin realizar una clasificación por el género de la película.

Finalmente, cada subtítulo que se descarga viene en un archivo con extensión .str, teniendo una estructura tal como se ve en la Figura 1:

```
1
00:01:09,459 --> 00:01:11,495
Sometimes, I think I'm cursed.

2
00:01:11,670 --> 00:01:14,457
Because of something that happened
before I was even born.

3
00:01:15,465 --> 00:01:18,502
See, a long time ago,
there was this family.

4
00:01:19,845 --> 00:01:21,710
The papa, he was a musician.

5
00:01:22,723 --> 00:01:26,341
He and his family would sing
and dance and count their blessings.
```

Fig. 4. Subtítulos de película

Cada uno de estos documentos con extensión .str maneja elementos que no son de interés para este análisis, por lo que antes de usarlos en el sistema se les hace una limpieza para eliminar cualquier carácter que no sea una letra y se guarda un nuevo archivo con extensión .txt. Esto se explicará más a detalle en el siguiente apartado, sin embargo el resultado obtenido es como el siguiente:

”once in a while when i wake up
i find myself crying
the dream i must have had
i can never recall
but
but
the sensation that ive lost something
lingers for a long time after i wake up
im always searching
for something for someone
this feeling has possessed me
i think from that day
that day when the stars came falling
it was almost as if
as if a scene from a dream
nothing more nothing less
than a beautiful view”

B. Preprocesamiento de los datos

Como se muestra en la Figura 1, los subtítulos se encuentran con algunos caracteres o datos que no son útiles para el objetivo de la investigación, por lo que se desarrolló un programa que elimina estos caracteres y que además, coloca en un solo archivo los subtítulos de todas las películas de cada clase, pasando de 400 archivos con extensión .srt a 4 archivos .txt correspondientes a cada clase, separada cada película por una marca de texto (—-NewMovie——) como se muestra en el ejemplo:

```
just being my friend
if he thinks someone knows me
hell kill them
the police dont go in there
they go after me not him
ive been in jail three times
while he goes on
killing undisturbed
and they dont arrest him
knockout ned
says the war will continue
police chief jose guedes
has promised to arrest
both gangs in the city of god
-----NewMovie-----
sometimes i think im cursed
because of something that happened
before i was even born
see a long time ago
there was this family
the papa he was a musician
he and his family would sing
and dance and count their blessings
but he also had a dream
to play for the world
and one day he left with his guitar
and never returned
and the mamé
she didnt have time to cry
over that walk-away musician
```

Fig. 5. Subtítulos con separadores

La división entre cada una de las películas se realizó porque resulta de utilidad para hacer el conteo de la cantidad de documentos totales con los que está operando en el sistema, y dicho valor es aplicado en el proceso de construcción de la tabla IDF.

C. Construcción del modelo

En este apartado se explica paso a paso el flujo que sigue el software para obtener la predicción de una película así como el proceso para entrenarlo. Para facilitar la explicación se utilizará la siguiente imagen:

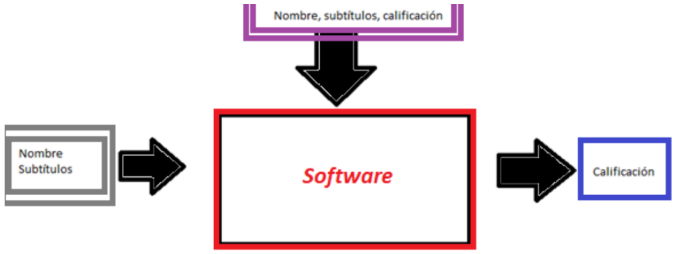

Fig. 6. Descripción del modelo

En primera instancia, el recuadro de color morado hace referencia a todos los archivos necesarios para entrenar el sistema, inicialmente se tenían estipulados 100 para cada categoría dando un total de 400. En el apartado anterior se explicó con detenimiento cuál fue el tratamiento que se realizó en ellos para que fueran aceptados por el modelo predictor, por lo que no se pretende abundar más en este tema. Ahora bien, el recuadro gris representa al archivo o archivos que se desean analizar. Hay dos opciones para estos documentos, puede limpiarse de manera manual con el código que trae el programa (que generaría un archivo .txt) o usar un archivo .srt (que por lo general es la extensión que traen al descargarlos de la página opensubtitles.com) y dejar que la aplicación realice todo el proceso automáticamente. Básicamente el usuario se encuentra aquí con la primera funcionalidad del sistema: limpieza de los archivos para entrenar o analizar. Cabe añadir que la funcionalidad prácticamente es la misma en ambos limpiadores, sólo que uno no inserta la leyenda "—–NewMovie——" ni mucho menos analiza una carpeta por completo y obtiene los documentos con una extensión en particular, simplemente trabaja con el documento que se le indique.

Por otra parte el recuadro azul es una cadena de texto, y sólo hay cuatro posibilidades: Mala, Regular, Buena o Excelente; ninguna otra más.

Finalmente, el recuadro central de color rojo es el motivo de esta sección, por lo que a continuación se describen cada uno de los componentes o módulos con los que cuenta ya que como tal reflejan el método que se utilizó durante este trabajo de investigación. Sin embargo, de forma muy general, se describe a continuación:

- Limpieza de cada uno de los archivos que conforman el dataset de entrenamiento
- Tokenización
- Stemming
- Obtención de las frecuencias de las palabras de cada documento
- Construcción de la matriz de TF
- Construcción de la matriz IDF
- Construcción de la matriz TF-IDF

Ahora, el proceso o método que sigue una película para ser analizada es muy similar, sólo que se realizan más pasos o menos de acuerdo al algoritmo. Inicialmente se pensó en trabajar con uno solo, sin embargo se presentaron un par

de cuestiones que como se explicará más adelante, derivaron la implementación de otros dos para dar sustento o justificación a nuestros resultados y comprobar que el algoritmo de nuestra autoría funciona adecuadamente. Pues bien, para el clasificador Bayesiano únicamente se requiere que el archivo a analizar sea limpiado y tokenizado, del resto se encarga el software; por otra parte los algoritmos de SVM y K-Neighbors requieren seguir todo el proceso desde el paso A hasta el G, con un paso extra que consiste en la transposición de la matriz TF-IDF resultante para que las columnas sean las palabras y las filas sean el archivo a analizar.

A continuación se describen cada uno de los pasos, y si es que hay, sus fórmulas asociadas. Además vale la pena mencionar que las herramientas principales para el desarrollo del sistema son dos: NLTK y Sklearn; ya que ambas bibliotecas están disponibles para Python y contienen los algoritmos para desarrollar todos los pasos.

1) Matriz de ocurrencia por películas: Dando inicio a la formación del modelo de predicción es necesario la construcción de una matriz que registre las ocurrencias de todas las palabras por cada guión de las películas, resultando de forma similar a la siguiente Figura:


```
{'\nThose Who Are ': {'resili': 1, 'stay': 1, 'game': 1, 'longer': 1,
'"': 1, 'mountain': 1}, 'However, I real': {'howev': 1, ',': 2,
'realis': 1, 'mani': 1, 'year': 1}, 'Have you experi': {'experienc':
1, 'thi': 1, 'befor': 1, '?': 1}, 'To be honest, I': {'honest': 1,
',': 1, '’': 1, 'answer': 1, '.': 1}, 'I can’t tell yo': {'’': 1,
'tell': 1, 'right': 1, 'cours': 1, 'action': 1, ';': 1, 'onli': 1,
'know': 1, '.': 1}...}
```


Fig. 7. Fragmento matriz de ocurrencias

2) Matriz TF: Posteriormente se necesita encontrar la frecuencia de término para cada palabra en un guión de una película. Para ello se aplica la siguiente fórmula:

TF (t) = (Número de veces que el término t aparece en un documento) / (Número total de términos en el documento)

Aquí, el documento es el guión de cada película (tomando en este momento gran importancia la división añadida en la preparación de los datos, lo cual nos permite saber el número de documentos y su delimitación),y a su vez el término es una palabra en un guión.

La matriz TF resulta similar a la siguiente Figura:


```
{'\nThose Who Are ': {'resili': 0.03225806451612903, 'stay':
0.03225806451612903, 'game': 0.03225806451612903, 'longer':
0.03225806451612903, '"': 0.03225806451612903, 'mountain':
0.03225806451612903}, 'However, I real': {'howev':
0.07142857142857142, ',': 0.14285714285714285, 'realis':
0.07142857142857142, 'mani': 0.07142857142857142, 'year':
0.07142857142857142}, 'Have you experi': {'experienc': 0.25, 'thi':
0.25, 'befor': 0.25, '?': 0.25}, 'To be honest, I': {'honest': 0.2,
',': 0.2, '’': 0.2, 'answer': 0.2, '.': 0.2}, 'I can’t tell yo':
{'’': 0.1111111111111111, 'tell': 0.1111111111111111, 'right':
0.1111111111111111, 'cours': 0.1111111111111111, 'action':
0.1111111111111111, ';': 0.1111111111111111, 'onli':
0.1111111111111111, 'know': 0.1111111111111111, '.':
0.1111111111111111}}
```


Fig. 8. Fragmento matriz TF

3) Palabras por Documento: Esta matriz se forma a partir del conteo de las veces que un conjunto de documentos contiene una palabra, lo que será útil para la próxima construcción de la matriz IDF, siendo el resultado similar a la siguiente figura:

Es decir, la palabra _____ aparece en _____ guiones.


```
{'resili': 2, 'stay': 2, 'game': 3, 'longer': 2, '"': 5, 'mountain':
1, 'truth': 1, 'never': 2, 'climb': 1, 'vain': 1, ':': 8, 'either':
1, 'reach': 1, 'point': 2, 'higher': 1, 'today': 1, ',': 22, 'train':
1, 'power': 4, 'abl': 1, 'tomorrow.': 1, '”': 5, '—': 3, 'friedrich':
1, 'nietzsch': 1, 'challeng': 2, 'setback': 2, 'meant': 1, 'defeat':
3, 'promot': 1, '.': 45, 'howev': 2, 'realis': 2, 'mani': 3, 'year':
4, 'crush': 1, 'spirit': 1, 'easier': 1, 'give': 4, 'risk': 1}
```


Fig. 9. Fragmento matriz de Palabras por documento

4) Matríz IDF: La tabla anterior es útil para la construcción de la matriz IDF, que como se hace mención en la introducción, se define como:

IDF (t) = log_e (Número total de documentos / Número de documentos con el término t)

De igual manera, un documento es cada guión de una película y el término es una palabra en un guión. La matriz IDF resulta similar a la siguiente Figura:


```
{'\nThose Who Are ': {'resili': 1.414973347970818, 'stay':
1.414973347970818, 'game': 1.2388820889151366, 'longer':
1.414973347970818, '"': 1.0170333392987803, 'mountain':
1.7160033436347992}, 'However, I real': {'howev': 1.414973347970818,
',': 0.37358066281259295, 'realis': 1.414973347970818, 'mani':
1.2388820889151366, 'year': 1.1139433523068367}, 'Have you experi':
{'experienc': 1.7160033436347992, 'thi': 1.1139433523068367, 'befor':
1.414973347970818, '?': 0.9378520932511555}, 'To be honest, I':
{'honest': 1.7160033436347992, ',': 0.37358066281259295, '’':
0.5118833609788743, 'answer': 1.414973347970818, '.':
0.06279082985945544}, 'I can’t tell yo': {'’': 0.5118833609788743,
'tell': 1.414973347970818, 'right': 1.1139433523068367, 'cours':
1.7160033436347992, 'action': 1.2388820889151366, ';':
1.7160033436347992, 'onli': 1.2388820889151366, 'know':
1.0170333392987803, '.': 0.06279082985945544}}
```


Fig. 10. Fragmento matriz IDF

5) Matriz TFIDF: Finalmente con las matrices IDF y TF construidas, se puede construir la nueva matriz TFIDF, que

en términos generales se multiplican los valores de ambas matrices, resultando una nueva por cada categoría similar a la siguiente:


```
{'\nThose Who Are ': {'resili': 0.04564430154744574, 'stay':
0.04564430154744574, 'game': 0.03996393835210118, 'longer':
0.04564430154744574, '"': 0.0328075270741542, 'mountain':
0.05535494656886449}, 'However, I real': {'howev':
0.10106952485505842, ',': 0.053368666116084706, 'realis':
0.10106952485505842, 'mani': 0.08849157777965261, 'year':
0.079567382307631 19}, 'Have you experi': {'experienc':
0.4290008359086998, 'thi': 0.2784858380767092, 'befor':
0.3537433369927045, '?': 0.2344630233 1278887}, 'To be honest, I':
{'honest': 0.34320066872695987, ',': 0.07471613256251859, '’':
0.10237667219577487, 'answer': 0.2829946695941636, '.':
0.01255816597189109}, 'I can’t tell yo': {'’': 0.0568759289976527,
'tell': 0.15721926088564644, 'right': 0.12377148358964851, 'cours':
0.19066703818164435, 'action': 0.13765356543501517, ';':
0.19066703818164435, 'onli': 0.13765356543501517, 'know':
0.11300370436653114, '.': 0.006976758873272827}}
```


Fig. 11. Fragmento matriz TF-IDF

Básicamente esta matriz sirve para contar el número de apariciones de palabra en el texto lo que posteriormente nos ayudará a calcular la importancia de éstas en el guión.

D. Modelo Predictivo

El proceso anterior se realiza con cada una de las 4 categorías, resultando 4 diferentes matrices TFIDF, las cuales son guardadas en un archivo con extensión CSV, que será el modelo predictivo que servirá como insumo para el funcionamiento de los clasificadores a implementar y de esta manera se evita tener que construir la matriz completa desde cero cada que inicie el programa. El archivo CSV se puede apreciar en la siguiente Figura:

Palabras	Malas	Regulares	Buenas	Excelentes
bibl	0.00076735	0.00102714	0.00085325	0.00210016
biblia	0.02141582	0	0	0
biblic	0.00139665	0.00151172	0	0
bibliographi	0.00263158	0	0	0
bic	0	0.00143781	0	0
bicarb	0	0.00249116	0.00136147	0
bicarbon	0	0.001221	0	0.00133251
bick	0	0	0	0.00368148
bicker	0.00209424	0.00059335	0.00177778	0.00126843
bickerin	0	0	0.00251256	0
bickl	0	0	0	0.00348556
bicycl	0.00216429	0.00146101	0.00070469	0.00178657
bid	0.00823506	0.00113337	0.00195243	0.00080519
bidden	0	0	0.00227015	0.00132628
bidder	0.00138822	0.00196271	0.00097752	0
biddl	0	0	0.00163532	0
bide	0.00188857	0	0.00126694	0.00107296
bidori	0	0	0.00178094	0

Fig. 12. Matriz TFIDF del modelo

E. Clasificadores

Los clasificadores son algoritmos que se utilizan para asignar un elemento entrante no etiquetado en una categoría concreta conocida. Dichos algoritmos, permiten, ordenar o disponer por clases elementos entrantes, a partir de cierta información característica de estos. Una manera de implementar un clasificador es seleccionar un conjunto de ejemplos etiquetados y tratar de definir una regla que permita asignar una etiqueta a cualquier otro dato de entrada. Al inicio del presente proyecto, se tenía en cuenta implementar un sólo clasificador, sin embargo, debido a los resultados negativos de éste, se decidió implementar 2 alternativas adicionales para mejorar o confirmar los resultados obtenidos. A continuación se describe su construcción y funcionamiento:

1) Clasificador Bayesiano: Este clasificador calcula la probabilidad que tiene una película de ser clasificada como mala, regular, buena o excelente a partir de su guión. Toma como entrada el modelo de la matriz TFIDF previamente creada, las probabilidades prior (0.25) de cada clase, el tamaño del vocabulario de cada clase, las probabilidades condicionales usados por el clasificador y la suma de los pesos de la matriz TF-IDF de todas las palabras del vocabulario completo. Las probabilidad condicional de cada palabra se dio a partir de una clase dada de la siguiente manera:

$$P(Palabra|Clase) = \frac{TF-IDF(Palabra, Clase)+1}{TF-IDF(Clase)+|V|} \tag{7}$$

Siendo $|V|$ el tamaño del vocabulario. Una vez finalizado la construcción del clasificador bayesiano, se procedió a obtener la predicción de la clases a partir de éste.

2) Clasificador de Laplace: En este clasificador se emplea Laplace y una escala logarítmica porque muchas veces las probabilidades serán tan pequeñas que en otra escala podrían provocar pérdida de información, evitando así esta problemática. El clasificador de Laplace recibe la tabla TF-IDF del modelo creado, una lista con las palabras del guión de la película a analizar, una constante K para aplanar la curva (siendo 1 en este estudio) y las clases disponibles para categorizar. Finalmente, una vez finalizado la construcción del clasificador de Laplace, se procedió a obtener la predicción de la clases a partir de éste.

3) Clasificador SVM: El clasificador por máquina de vectores soporte recibe la tabla TF-IDF del modelo y la tabla TF-IDF de la película a analizar, es importante señalar que este algoritmo cuenta con varios tipos de kernel. Puede ser ’linear’, ’poly’, ’rbf’, ’sigmoid’ o ’precomputed’; y se implementó el algoritmo usando la primera para nuestro estudio. Una vez finalizado la construcción del clasificador de SVM, se procedió a obtener la predicción de la clases a partir de éste.

4) Clasificador KNeighbors: Este clasificador utiliza al igual que lo anteriores la matriz TF-IDF construida para el modelo, el número de vecinos del algoritmo (1 a 3 en este estudio) así como tabla TF-IDF de la película a analizar.

Una vez teniendo construidos todos los clasificadores, el sistema realiza la evaluación de las películas a analizar con cada uno de éstos describiendo los resultados en la siguiente sección.

F. Descripción del experimento

Inicialmente se tenía previsto realizar el experimento con 20 documentos distintos para probar la funcionalidad del algoritmo de nuestra autoría, sin embargo los resultados realmente no aseguraban que el clasificador de Laplace estuviera haciendo su proceso de manera adecuada, lo que derivó la implementación de otros algoritmos más y la variación de cantidad de documentos con los que se construye la matriz TF-IDF para entrenar el sistema, de esta manera es posible visualizar la eficacia de cada matriz construida con N cantidad de documentos y la posibilidad de comparar los resultados obtenidos por nuestro algoritmo contra los otros dos añadidos que ya fueron importados desde las bibliotecas de NLTK y SKLearn de Python.

Ahora bien, dicho esto es posible describir a fondo en qué consistió el experimento. Antes que todo, la primera problemática que se abordó fue la de asegurar que Laplace estuviera trabajando correctamente, y como recientemente se mencionó, se añadieron dos nuevos algoritmos que son SVM (Máquina de Vectores de Soporte) y K-Neighbors para subsanar este problema.

Posteriormente se realizó la variación de la cantidad de documentos que se están usando para entrenar el sistema, por lo que la máxima cantidad eran 100 por categoría y a partir de ahí se restaban 10 cada vez que se generaba la tabla de entrenamiento hasta llegar a 30 películas por categoría como mínimo. Los criterios para seleccionar los archivos que se suprimen del dataset básicamente son dos: para las películas excelentes y buenas se van quitando los documentos con mayor cantidad de información (reflejado directamente en el tamaño del archivo) y el segundo criterio es que para las películas regulares y malas se quitan los archivos que tengan menor peso.

Debido a la cantidad tan alta de datos que maneja la computadora en cada ejecución, se decidió usar 8 películas de prueba (2 por cada categoría).

Una vez explicada la manera en que se distribuyen las películas durante el experimento, se procede a describir cómo es el flujo de éste. Para cada cantidad de películas desde 30 a 100 para cada categoría, se hace la limpieza de cada una de ellas y se generan cuatro archivos (uno de películas malas, otro de regulares, posteriormente uno de buenas y finalmente uno de excelentes), por consiguiente se crea la tabla de entrenamiento para cada uno de los algoritmos y se procede a entrenarlo. Ahora bien, cada una de las ocho películas seleccionadas se limpia y se le manda como parámetro a cada uno de los algoritmos para que comience a operar con esos datos. Finalmente se obtiene una cadena de texto referente al nivel de aceptación que tendrá.

IV. Resultados

A continuación se muestran los resultados obtenidos a partir de una cantidad N de documentos para entrenar el sistema y su relación con el número de veces que acertaron correctamente los algoritmos con las películas de prueba:

Cantidad de documentos de entrenamiento por categoría	Laplace	SVM	N-vecinos
100	1	2	2
90	2	2	2
80	1	2	2
70	1	2	2
60	2	1	2
50	2	2	2
40	2	2	2
30	1	2	2
20	1	2	2

Fig. 13. Resultados

Gráficamente los resultados pueden ser vistos de la siguiente manera:

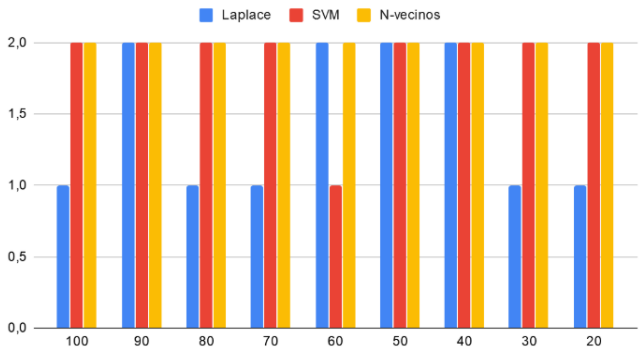

Fig. 14. Gráfica de resultados

Donde el eje X corresponde a la cantidad de películas utilizadas, es decir 100 por categoría, 90, 80 y así sucesivamente hasta llegar a 20. Ahora bien, el eje de las ordenadas describe la cantidad de veces que acertaron correctamente los algoritmos, dicho de otro modo refleja cuántas veces hicieron una buena predicción, y como se observa, la máxima cantidad de veces que lograron hacerlo fue 2. Sin embargo, para poder entender dichos resultados es necesario recurrir a las tablas que originaron estos gráficos.

Cantidad de documentos para entrenar cada categoría:	100			
Nombre Pelicula	**Laplace**	**SVM**	**N-vecinos**	**Real**
Dunkirk	Excelente	Mala	Buena	Buena
Jim Carrey The Unnatural	Excelente	Mala	Buena	Buena
Banking on africa the bitcoin revolution	Buena	Mala	Buena	Excelente
Marvels The punisher	Excelente	Mala	Buena	Excelente
2_English	Excelente	Mala	Buena	Mala
Skinned	Excelente	Mala	Buena	Mala
Splice	Excelente	Mala	Buena	Regular
Stranded	Buena	Mala	Buena	Regular

Fig. 15. 10 películas

Cantidad de documentos para entrenar:	90			
Nombre Pelicula	**Laplace**	**SVM**	**N-vecinos**	**Real**
Dunkirk	Mala	Buena	Buena	Buena
Jim Carrey The Unnatural	Mala	Buena	Buena	Buena
Banking on africa the bitcoin revolution	Mala	Buena	Buena	Excelente
Marvels The punisher	Mala	Buena	Buena	Excelente
2_English	Mala	Buena	Buena	Mala
Skinned	Mala	Buena	Buena	Mala
Splice	Mala	Buena	Buena	Regular
Stranded	Mala	Buena	Buena	Regular

Fig. 16. 90 películas

Cantidad de documentos para entrenar:	80			
NombrePelicula	**Laplace**	**SVM**	**N-vecinos**	**Real**
Dunkirk	Mala	Excelente	Buena	Buena
Jim Carrey The Unnatural	Mala	Excelente	Buena	Buena
Banking on africa the bitcoin revolution	Regular	Excelente	Buena	Excelente
Marvels The punisher	Mala	Excelente	Buena	Excelente
2_English	Regular	Excelente	Buena	Mala
Skinned	Regular	Excelente	Buena	Mala
Splice	Mala	Excelente	Buena	Regular
Stranded	Regular	Excelente	Buena	Regular

Fig. 17. 80 películas

Cantidad de documentos para entrenar:	70			
NombrePelicula	**Laplace**	**SVM**	**N-vecinos**	**Real**
Dunkirk	Excelente	Buena	Buena	Buena
Jim Carrey The Unnatural	Regular	Buena	Buena	Buena
Banking on africa the bitcoin revolution	Regular	Buena	Buena	Excelente
Marvels The punisher	Regular	Buena	Buena	Excelente
2_English	Regular	Buena	Buena	Mala
Skinned	Regular	Buena	Buena	Mala
Splice	Excelente	Buena	Buena	Regular
Stranded	Regular	Buena	Buena	Regular

Fig. 18. 70 películas

Cantidad de documentos para entrenar:	60			
NombrePelicula	**Laplace**	**SVM**	**N-vecinos**	**Real**
Dunkirk	Buena	Regular	Excelente	Buena
Jim Carrey The Unnatural	Buena	Regular	Excelente	Buena
Banking on africa the bitcoin revolution	Buena	Regular	Excelente	Excelente
Marvels The punisher	Buena	Regular	Excelente	Excelente
2_English	Buena	Regular	Excelente	Mala
Skinned	Buena	Regular	Excelente	Mala
Splice	Buena	Regular	Excelente	Regular
Stranded	Buena	Regular	Excelente	Regular

Fig. 19. 60 películas

Cantidad de documentos para entrenar:	50			
NombrePelicula	**Laplace**	**SVM**	**N-vecinos**	**Real**
Dunkirk	Buena	Mala	Buena	Buena
Jim Carrey The Unnatural	Buena	Mala	Buena	Buena
Banking on africa the bitcoin revolution	Buena	Mala	Buena	Excelente
Marvels The punisher	Buena	Mala	Buena	Excelente
2_English	Buena	Mala	Buena	Mala
Skinned	Buena	Mala	Buena	Mala
Splice	Buena	Mala	Buena	Regular
Stranded	Buena	Mala	Buena	Regular

Fig. 20. 50 películas

Cantidad de documetos para entrenar:	40			
NombrePelicula	**Laplace**	**SVM**	**N-vecinos**	**Real**
Dunkirk	Mala	Mala	Buena	Buena
Jim Carrey The Unnatural	Mala	Mala	Buena	Buena
Banking on africa the bitcoin revolution	Mala	Mala	Buena	Excelente
Marvels The punisher	Mala	Mala	Buena	Excelente
2_English	Mala	Mala	Buena	Mala
Skinned	Mala	Mala	Buena	Mala
Splice	Mala	Mala	Buena	Regular
Stranded	Mala	Mala	Buena	Regular

Fig. 21. 40 películas

Cantidad de documentos para entrenar:	30			
NombrePelicula	**Laplace**	**SVM**	**N-vecinos**	**Real**
Dunkirk	Mala	Regular	Buena	Buena
Jim Carrey The Unnatural	Regular	Regular	Buena	Buena
Banking on africa the bitcoin revolution	Regular	regular	Buena	Excelente
Marvels The punisher	Mala	Regular	Buena	Excelente
2_English	Mala	Regular	Buena	Mala
Skinned	Regular	Regular	Buena	Mala
Splice	Mala	Regular	Buena	Regular
Stranded	Mala	Regular	Buena	Regular

Fig. 22. 30 películas

Cantidad de documentos para entrenar:	20			
NombrePelicula	Laplace	SVM	N-vecinos	Real
Dunkirk	Mala	Buena	Buena	Buena
Jim Carrey The Unnatural	Mala	Buena	Buena	Buena
Banking on africa the bitcoin revolution	Mala	Buena	Buena	Excelente
Marvels The punisher	Mala	Buena	Buena	Excelente
2_English	Regular	Buena	Buena	Mala
Skinned	Mala	Buena	Buena	Mala
Splice	Mala	Buena	Buena	Regular
Stranded	Mala	Buena	Buena	Regular

Fig. 23. 20 películas

La columna relacionada con el algoritmo de Laplace presenta variaciones en sus resultados, sin embargo también llega a devolver muchas veces un mismo valor, en consecuencia, la probabilidad de que retorne una respuesta correcta por lo menos en una película es alta pero eso no significa que dicho resultado provenga de una predicción como tal.

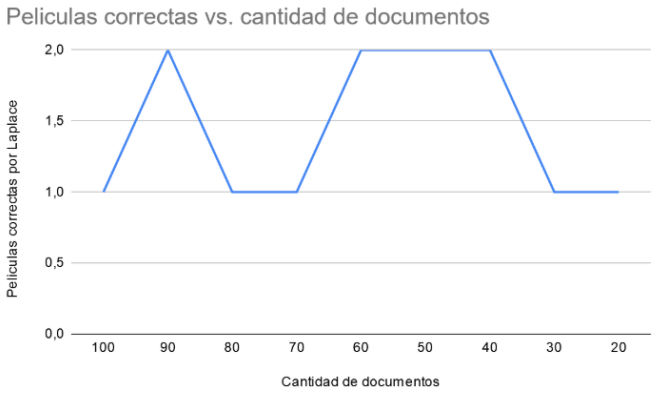

Fig. 24. Laplace

Para el algoritmo de SVM y N-neighbors no se destinan gráficos de manera particular porque tienden a devolver un mismo resultado con escasas variaciones para cada N cantidad de películas, lo que hace pensar que simplemente no son capaces de clasificar las películas, por lo menos con las condiciones implementadas en este trabajo.

Por otra parte, analizando el comportamiento de las tablas se puede ver una especie de separación entre las películas que caen en la categoría de malas y regulares con respecto a las buenas y excelentes, como si sólo hubiera esos dos grupos ya que en ninguna iteración del experimento se encontró una predicción excelente y una mala a la vez; a partir de esto se infiere que existe una relación entre la cantidad de documentos que se están manipulando para entrenar al sistema con respecto a la densidad de la curva, pues conforme disminuye la cantidad de documentos, el sistema comienza a predecir prácticamente toda entrada como mala. Para demostrar esto se realizó la gráfica categoría (o calificación) vs. cantidad de documentos.

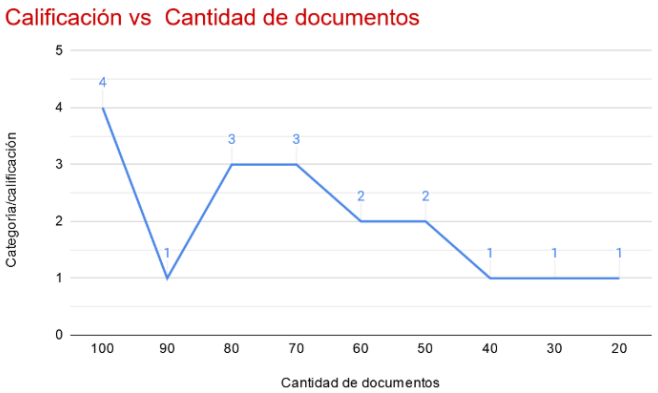

Fig. 25. Representación de resultados

La calificación 1 equivale a malas, 2 a regulares, 3 a buenas y 4 a excelentes. Ahora bien, cuando la cantidad de documentos es 100 hay un sobreentrenamiento del sistema, por lo que sólo es capaz de dar una evaluación correcta siempre y cuando conozca dicho documento, sin embargo la tendencia que marca el sistema es que clasifica todo prácticamente como excelente y es probable que tenga relación directa con la cantidad de información extra que tiene esta clase con respecto a las demás (50,000 líneas de texto, aproximadamente), luego clasifica mayoritariamente como malos a todos los documentos que se analizan (con 90 películas por categoría) y es a partir de este punto cuando se vuelve más notorio cómo al calcular las probabilidades del clasificador bayesiano simplemente se va desplazando la densidad de la curva hacia una categoría en particular, y eso explica por qué nunca se obtienen predicciones excelentes y malas en una misma prueba, pues simplemente es como si no tuviera mucha información de los extremos.

Finalmente, si se aplica el mismo proceso con los algoritmos de SVM y N-neighbors, y se enciman los resultados sobre la gráfica anterior se observa lo siguiente:

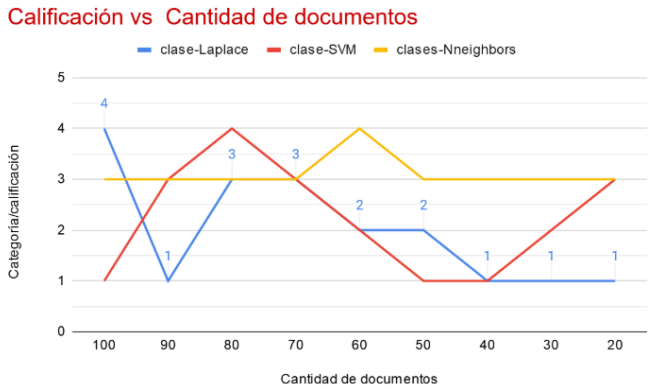

Fig. 26. Representación de resultados

El algoritmo SVM tiende a clasificar todo como malas, luego como buenas, posteriormente como excelentes y finalmente repite el proceso; en cambio N-neighbors prácticamente siempre clasifica como todo bueno a excepción de cuando se

usan 70 documentos. Este punto es crítico porque es el único momento en el que se intersectan los 3 algoritmos, por lo que se hizo una prueba más con 65 documentos para conocer el comportamiento pero con la diferencia que ahora se eliminaron 35 documentos de manera aleatoria y no como se definió en el experimento. Dicha prueba arrojó el siguiente resultado:

Cantidad de documentos para entrenar:	65			
NombrePelicula	Laplace	SVM	N-vecinos	Real
Dunkirk	Mala	Regular	Buena	Buena
Jim Carrey The Unnatural	Regular	Regular	Buena	Buena
Banking on africa the bitcoin revolution	Regular	Regular	Buena	Excelente
Marvels The punisher	Regular	Regular	Buena	Excelente
2_English	Regular	Regular	Buena	Mala
Skinned	Regular	Regular	Buena	Mala
Splice	Regular	Regular	Buena	Regular
Stranded	Mala	Regular	Buena	Regular

Fig. 27. Nueva Prueba

Prácticamente no hubo diferencia alguna, los tres clasificadores continuaron con la tendencia vista en la gráfica Calificación vs. Cantidad de documentos. Por último, también se puede encontrar al final de este documento un anexo con las evaluaciones del sistema a través de las medidas de precisión, recall y F1; sin embargo presenta resultados sumamente precarios. En el mejor de los casos el sistema tendrá 6.2% de precisión, 25% de recall (exhaustividad) y de 10% de F1.

V. Conclusiones

Después de haber realizado el experimento y obtener los resultados correspondientes se puede observar que en ocasiones el sistema retorna valores que llegan a ser similares a los reales, pero no son correctos en la mayoría de los casos. Si se observa desde la perspectiva de un director de cine lo que hasta ahora estamos procesando, elementos como la actuación, iluminación, música y ambientación, calidad de la imagen, sonido, originalidad o producción están pasando a segundo plano, y eso muestra que en el universo cinematográfico el guión una la película es tan solo una minúscula proporción de todo lo que se considera al evaluar un filme, por lo que es totalmente comprensible que el resultado real no se parezca al arrojado por el sistema. Sin embargo ¿Qué pasa cuando el sistema lanza una calificación de película igual a la que dan los críticos? Pues bien, es una pregunta que puede ser respondida o debatida desde distintas perspectivas, pues lo más lógico es pensar que los diálogos tienen más peso que cualquier otro elemento, lo cual también puede ser posible, sin embargo, con base en los resultados, es más probable que se deba a que dicha película cae en dónde está la mayor densidad de probabilidad de ese algoritmo, por lo que es complicado poder asegurar si la evaluación obtenida es confiable; hay que recordar que con base en los cálculos de la evaluación del sistema, se espera como mucho una precisión del 6.2% y un recall del 25%, es decir, qué tanto porcentaje de evaluaciones correctas podrá identificar el sistema. Estos datos son muy generales, mucho depende con cuántos y cuáles documentos se usen para entrenar al sistema pero por lo visto en el experimento, aún con estos cambios no se esperan resultados esperanzadores.

Por otro lado, para poder introducir la películas al sistema creamos al inicio un limpiador de texto que borraba todos lo caracteres especiales y dejaba sólo aquello que fueran letras, pero al principio este programa no quitaba los guiones medios y al pasar el texto al clasificador con este símbolo, provocaba que los resultados se alajaran aún más de lo esperado, por lo que esta consideración vale la pena tenerla presente al desarrollar este tipo de sistemas predictivos porque puede tener una gran influencia sobre los resultados y es mejor resolverlo desde el preprocesamiento.

Con respecto al clasificador como tal, hay una serie de cuestiones que sobresalen en este tema. En primer lugar, en general, los resultados no siempre son cercanos a los esperados, ya justificamos con evaluaciones al sistema qué tan bueno puede llegar a ser, sin embargo ahora se aspira a encontrar explicaciones. La primera de ellas se enfoca a la forma en que sesgamos la escala de IMDB, ya que se le asignó una gran parte a lo que consideramos como malas, por lo que a los algoritmos se les dificulta colocar límites entre las clases; también es importante introducir la variable de los géneros en el cine, ya que es muy posible que haya ciertos temas que estén más relacionados a una película mala que a una buena o excelente; por ejemplo, en el 2009 surgió un filme que se llamó Dragon Ball Evolution que fue sumamente criticada por los espectadores porque no cumplía con sus expectativas tanto visuales como de diálogos, sin embargo la película se basaba en el Animé de Dragon Ball que ha sido amado por muchas generaciones desde que se comercializó. Este es un ejemplo claro que no sólo el guión importa, también otros elementos están en juego. Una posible solución a este trabajo de investigación podría ser reformular y construir nuevamente el set de datos que se usarán para entrenar al sistema dividiendo por géneros las películas para entrenar los algoritmos de forma más justa.

Esto último desprende una observación importante con respecto a las películas que se utilizaron, puesto que a pesar de que las cuatro categorías cuentan con 100 guiones como máximo, al ordenarlas de mayor a menor por el tamaño de sus archivos, las mejor evaluadas llegan a tener más de 100 kb de información y conforme empiezan a tener una calificación menor, los diálogos cada vez son menores. Además se detecta una vulnerabilidad más en el sistema, puesto que al usar todos los archivos esa diferencia comienza a impactar considerablemente en la construcción de las tablas TF-IDF. Ahora bien, para entender las dimensiones de estas diferencias se resume de manera sencilla: una vez limpiadas las películas y puestas en un solo documento para cada clase

de la escala, las películas excelentes tienen un excedente de 50,000 líneas de texto por encima de las regulares y de las malas. Para subsanar este problema en el experimento se decidió ir eliminando los archivos con mayor texto en las películas excelentes y buenas, y quitar las que poseen menos información en el caso de las regulares y malas pero como se observó en los resultados, realmente no impacta tanto si se hace inclusive de manera aleatoria porque continúa habiendo diferencia entre la cantidad de información en cada clase, por lo que a fin de cuentas el patrón de comportamiento continuó siendo el mismo.

Pues bien, otra razón que provocó modificaciones sobre el proyecto inicial fue el sobreentrenamiento. Realmente gran parte del éxito de un sistema predictivo se relaciona con la elección correcta de los datos de entrenamiento, sin embargo tampoco hay parámetros claros sobre cómo elegir correctamente esta información; lo cierto es que sobrepasar una cantidad de datos impacta demasiado en el sistema porque llega un punto en el que está sobreentrenado y no es capaz de predecir nada que no pertenezca a los datos que ya conoce. Ahora bien, esto aunado a que a pesar de probar diferente cantidad de archivos no hizo un cambio considerable en la manera de operar (pues las gráficas indican que a pesar de tener menos documentos, los algoritmos no mostraron mejoría al momento de realizar los predicciones) llevó a considerar que quizás para el caso de las Máquinas de Vectores de Soporte es necesario cambiar el tipo de Kernel, ya que una línea recta posiblemente no es capaz de separar las clasificaciones pues las palabras contenidas en la matriz TF-IDF se presentan muchas veces en los cuatro casos (malas, regulares, buenas y excelentes).

En suma, el sistema desarrollado no es capaz de predecir si una película tendrá buena aceptación tanto por cuestiones probabilísticas como algorítmicas e inclusive de cualidades, pues regresando nuevamente al inicio de este apartado se mencionó que una película no necesariamente es catalogada como buena por el simple hecho de tocar un cierto tema o mencionar algunas palabras en particular. Cada año cuando se llevan a cabo los premios Oscar, por ejemplo, los filmes son nominados por diferentes razones, algunas sí son por el diálogo pero no es lo único visible, en otras su fuerte son los efectos especiales, otras más tienden al tracklist, en fin, la cantidad de variables que se manejan en este ámbito son tantas que muy posiblemente ningún sistema logre determinar una evaluación suficientemente satisfactoria a menos que se integren nuevos parámetros. Esto finalmente da apertura a que se desarrollen más trabajos a futuro, algunos de ellos son por ejemplo, introducir al sistema información acerca de los actores y características asociadas a su personalidad, su manera de actuar frente a una cámara, qué tan conocido son, entre otras más, sin embargo todas ellas sumamente difíciles de cuantificar. Una segunda posibilidad es determinar la cantidad correcta de películas que logren abarcar una cierta cantidad de géneros que sean representativos y aún mejor, que los filmes puedan potenciar la discriminación entre ellos y de esta manera se le facilitará a los algoritmos notar las diferencias entre cada una de las clases.

Finalmente, a pesar de que no fue posible generar una herramienta lo suficientemente eficaz para hacer predicciones confiables, logramos el objetivo de responder con las condiciones estipuladas en esta investigación la pregunta inicial, y lo más importante es que se hizo notar la necesidad de una amplia cantidad de investigaciones en el ámbito del procesamiento digital de textos y en el mundo cinematográfico.

VI. Anexo Matrices de Conufusión

Laplace con 100 Películas				
Actual/Predicted	Excelente	Buena	Regular	Mala
Excelente	1	1	0	0
Buena	2	0	0	0
Regular	1	1	0	0
Mala	2	0	0	0

Laplace con 90 Películas				
Actual/Predicted	Excelente	Buena	Regular	Mala
Excelente	0	0	0	2
Buena	0	0	0	2
Regular	0	0	0	2
Mala	0	0	0	2

Laplace con 80 Películas				
Actual/Predicted	Excelente	Buena	Regular	Mala
Excelente	0	0	1	1
Buena	0	0	0	2
Regular	0	0	1	1
Mala	0	0	2	0

Laplace con 70 Películas				
Actual/Predicted	Excelente	Buena	Regular	Mala
Excelente	0	0	2	0
Buena	1	0	1	0
Regular	1	0	1	0
Mala	0	0	2	0

Laplace con 60 Películas				
Actual/Predicted	Excelente	Buena	Regular	Mala
Excelente	0	2	0	0
Buena	0	2	0	0
Regular	0	2	0	0
Mala	0	2	0	0

Laplace con 50 Películas				
Actual/Predicted	Excelente	Buena	Regular	Mala
Excelente	0	2	0	0
Buena	0	2	0	0
Regular	0	2	0	0
Mala	0	2	0	0

Laplace con 40 Películas				
Actual/Predicted	Excelente	Buena	Regular	Mala
Excelente	0	0	0	2
Buena	0	0	0	2
Regular	0	0	0	2
Mala	0	0	0	2

Laplace con 30 Películas				
Actual/Predicted	Excelente	Buena	Regular	Mala
Excelente	0	0	1	1
Buena	0	0	1	1
Regular	0	0	0	2
Mala	0	0	1	1

Laplace con 20 Películas				
Actual/Predicted	Excelente	Buena	Regular	Mala
Excelente	0	0	0	2
Buena	0	0	0	2
Regular	0	0	0	2
Mala	0	0	1	1

SVM con 100 Películas				
Actual/Predicted	Excelente	Buena	Regular	Mala
Excelente	0	0	0	2
Buena	0	0	0	2
Regular	0	0	0	2
Mala	0	0	0	2

SVM con 90 Películas				
Actual/Predicted	Excelente	Buena	Regular	Mala
Excelente	0	2	0	0
Buena	0	2	0	0
Regular	0	2	0	0
Mala	0	2	0	0

SVM con 80 Películas				
Actual/Predicted	Excelente	Buena	Regular	Mala
Excelente	2	0	0	0
Buena	2	0	0	0
Regular	2	0	0	0
Mala	2	0	0	0

SVM con 70 Películas				
Actual/Predicted	Excelente	Buena	Regular	Mala
Excelente	0	2	0	0
Buena	0	2	0	0
Regular	0	2	0	0
Mala	0	2	0	0

SVM con 60 Películas				
Actual/Predicted	Excelente	Buena	Regular	Mala
Excelente	0	0	2	0
Buena	0	0	2	0
Regular	0	0	2	0
Mala	0	0	2	0

SVM con 50 Películas				
Actual/Predicted	Excelente	Buena	Regular	Mala
Excelente	0	0	0	2
Buena	0	0	0	2
Regular	0	0	0	2
Mala	0	0	0	2

SVM con 40 Películas				
Actual/Predicted	Excelente	Buena	Regular	Mala
Excelente	0	0	0	2
Buena	0	0	0	2
Regular	0	0	0	2
Mala	0	0	0	2

SVM con 30 Películas				
Actual/Predicted	Excelente	Buena	Regular	Mala
Excelente	0	0	2	0
Buena	0	0	2	0
Regular	0	0	2	0
Mala	0	0	2	0

SVM con 20 Películas				
Actual/Predicted	Excelente	Buena	Regular	Mala
Excelente	0	2	0	0
Buena	0	2	0	0
Regular	0	2	0	0
Mala	0	2	0	0

N-Vecinos con 100 Películas				
Actual/Predicted	Excelente	Buena	Regular	Mala
Excelente	0	2	0	0
Buena	0	2	0	0
Regular	0	2	0	0
Mala	0	2	0	0
N-Vecinos con 90 Películas				
Actual/Predicted	Excelente	Buena	Regular	Mala
Excelente	0	2	0	0
Buena	0	2	0	0
Regular	0	2	0	0
Mala	0	2	0	0
N-Vecinos con 80 Películas				
Actual/Predicted	Excelente	Buena	Regular	Mala
Excelente	0	2	0	0
Buena	0	2	0	0
Regular	0	2	0	0
Mala	0	2	0	0
N-Vecinos con 70 Películas				
Actual/Predicted	Excelente	Buena	Regular	Mala
Excelente	0	2	0	0
Buena	0	2	0	0
Regular	0	2	0	0
Mala	0	2	0	0
N-Vecinos con 60 Películas				
Actual/Predicted	Excelente	Buena	Regular	Mala
Excelente	2	0	0	0
Buena	2	0	0	0
Regular	2	0	0	0
Mala	2	0	0	0
N-Vecinos con 50 Películas				
Actual/Predicted	Excelente	Buena	Regular	Mala
Excelente	0	2	0	0
Buena	0	2	0	0
Regular	0	2	0	0
Mala	0	2	0	0
N-Vecinos con 40 Películas				
Actual/Predicted	Excelente	Buena	Regular	Mala
Excelente	0	2	0	0
Buena	0	2	0	0
Regular	0	2	0	0
Mala	0	2	0	0
N-Vecinos con 30 Películas				
Actual/Predicted	Excelente	Buena	Regular	Mala
Excelente	0	2	0	0
Buena	0	2	0	0
Regular	0	2	0	0
Mala	0	2	0	0
N-Vecinos con 20 Películas				
Actual/Predicted	Excelente	Buena	Regular	Mala
Excelente	0	2	0	0
Buena	0	2	0	0
Regular	0	2	0	0
Mala	0	2	0	0

VII. Anexo De Resultados del Modelo

Laplace con 100 Películas				
	precision	recall	f1-score	support
1 Excelente	0.167	0.500	0.250	2
2 Buena	0.000	0.000	0.000	2
3 Mala	0.000	0.000	0.000	2
4 Regular	0.000	0.000	0.000	2
avg / total	0.042	0.125	0.062	8
Laplace con 90 Películas				
	precision	recall	f1-score	support
1 Excelente	0.000	0.000	0.000	2
2 Buena	0.000	0.000	0.000	2
3 Mala	0.000	0.000	0.000	2
4 Regular	0.250	1.000	0.400	2
avg / total	0.062	0.250	0.100	8
Laplace con 80 Películas				
	precision	recall	f1-score	support
1 Excelente	0.000	0.000	0.000	2
2 Buena	0.000	0.000	0.000	2
3 Mala	0.250	0.500	0.333	2
4 Regular	0.000	0.000	0.000	2
avg / total	0.062	0.125	0.083	8
Laplace con 70 Películas				
	precision	recall	f1-score	support
1 Excelente	0.000	0.000	0.000	2
2 Buena	0.000	0.000	0.000	2
3 Mala	0.167	0.500	0.250	2
4 Regular	0.000	0.000	0.000	2
avg / total	0.042	0.125	0.062	8
Laplace con 60 Películas				
	precision	recall	f1-score	support
1 Excelente	0.000	0.000	0.000	2
2 Buena	0.250	1.000	0.400	2
3 Mala	0.000	0.000	0.000	2
4 Regular	0.000	0.000	0.000	2
avg / total	0.062	0.250	0.100	8
Laplace con 50 Películas				
	precision	recall	f1-score	support
1 Excelente	0.000	0.000	0.000	2
2 Buena	0.250	1.000	0.400	2
3 Mala	0.000	0.000	0.000	2
4 Regular	0.000	0.000	0.000	2
avg / total	0.062	0.250	0.100	8

Laplace con 40 Películas	precision	recall	f1-score	support
1 Excelente	0.000	0.000	0.000	2
2 Buena	0.000	0.000	0.000	2
3 Mala	0.000	0.000	0.000	2
4 Regular	0.250	1.000	0.400	2
avg / total	0.062	0.250	0.100	8

Laplace con 30 Películas	precision	recall	f1-score	support
1 Excelente	0.000	0.000	0.000	2
2 Buena	0.000	0.000	0.000	2
3 Mala	0.000	0.000	0.000	2
4 Regular	0.200	0.500	0.286	2
avg / total	0.050	0.125	0.071	8

Laplace con 20 Películas	precision	recall	f1-score	support
1 Excelente	0.000	0.000	0.000	2
2 Buena	0.000	0.000	0.000	2
3 Mala	0.000	0.000	0.000	2
4 Regular	0.143	0.500	0.222	2
avg / total	0.036	0.125	0.056	8

SVM con 100 Películas	precision	recall	f1-score	support
1 Excelente	0.000	0.000	0.000	2
2 Buena	0.000	0.000	0.000	2
3 Mala	0.000	0.000	0.000	2
4 Regular	0.250	1.000	0.400	2

e

SVM con 90 Películas	precision	recall	f1-score	support
1 Excelente	0.000	0.000	0.000	2
2 Buena	0.250	1.000	0.400	2
3 Mala	0.000	0.000	0.000	2
4 Regular	0.000	0.000	0.000	2
avg / total	0.062	0.250	0.100	8

SVM con 80 Películas	precision	recall	f1-score	support
1 Excelente	0.250	1.000	0.400	2
2 Buena	0.000	0.000	0.000	2
3 Mala	0.000	0.000	0.000	2
4 Regular	0.000	0.000	0.000	2
avg / total	0.062	0.250	0.100	8

SVM con 70 Películas	precision	recall	f1-score	support
1 Excelente	0.000	0.000	0.000	2
2 Buena	0.250	1.000	0.400	2
3 Mala	0.000	0.000	0.000	2
4 Regular	0.000	0.000	0.000	2
avg / total	0.062	0.250	0.100	8

SVM con 60 Películas	precision	recall	f1-score	support
1 Excelente	0.000	0.000	0.000	2
2 Buena	0.000	0.000	0.000	2
3 Mala	0.250	1.000	0.400	2
4 Regular	0.000	0.000	0.000	2
avg / total	0.062	0.250	0.100	8

SVM con 50 Películas	precision	recall	f1-score	support
1 Excelente	0.000	0.000	0.000	2
2 Buena	0.000	0.000	0.000	2
3 Mala	0.000	0.000	0.000	2
4 Regular	0.250	1.000	0.400	2
avg / total	0.062	0.250	0.100	8

SVM con 40 Películas	precision	recall	f1-score	support
1 Excelente	0.000	0.000	0.000	2
2 Buena	0.000	0.000	0.000	2
3 Mala	0.000	0.000	0.000	2
4 Regular	0.250	1.000	0.400	2
avg / total	0.062	0.250	0.100	8

SVM con 30 Películas	precision	recall	f1-score	support
1 Excelente	0.000	0.000	0.000	2
2 Buena	0.000	0.000	0.000	2
3 Mala	0.250	1.000	0.400	2
4 Regular	0.000	0.000	0.000	2
avg / total	0.062	0.250	0.100	8

SVM con 20 Películas	precision	recall	f1-score	support
1 Excelente	0.000	0.000	0.000	2
2 Buena	0.250	1.000	0.400	2
3 Mala	0.000	0.000	0.000	2
4 Regular	0.000	0.000	0.000	2
avg / total	0.062	0.250	0.100	8

N-Vecinos con 100 Películas	precision	recall	f1-score	support
1 Excelente	0.000	0.000	0.000	2
2 Buena	0.250	1.000	0.400	2
3 Mala	0.000	0.000	0.000	2
4 Regular	0.000	0.000	0.000	2
avg / total	0.062	0.250	0.100	8

N-Vecinos con 90 Películas	precision	recall	f1-score	support
1 Excelente	0.000	0.000	0.000	2
2 Buena	0.250	1.000	0.400	2
3 Mala	0.000	0.000	0.000	2
4 Regular	0.000	0.000	0.000	2
avg / total	0.062	0.250	0.100	8

N-Vecinos con 80 Películas	precision	recall	f1-score	support
1 Excelente	0.000	0.000	0.000	2
2 Buena	0.250	1.000	0.400	2
3 Mala	0.000	0.000	0.000	2
4 Regular	0.000	0.000	0.000	2
avg / total	0.062	0.250	0.100	8

N-Vecinos con 70 Películas	precision	recall	f1-score	support
1 Excelente	0.000	0.000	0.000	2
2 Buena	0.250	1.000	0.400	2
3 Mala	0.000	0.000	0.000	2
4 Regular	0.000	0.000	0.000	2
avg / total	0.062	0.250	0.100	8

N-Vecinos con 60 Películas	precision	recall	f1-score	support
1 Excelente	0.250	1.000	0.400	2
2 Buena	0.000	0.000	0.000	2
3 Mala	0.000	0.000	0.000	2
4 Regular	0.000	0.000	0.000	2
avg / total	0.062	0.250	0.100	8

N-Vecinos con 50 Películas	precision	recall	f1-score	support
1 Excelente	0.000	0.000	0.000	2
2 Buena	0.250	1.000	0.400	2
3 Mala	0.000	0.000	0.000	2
4 Regular	0.000	0.000	0.000	2
avg / total	0.062	0.250	0.100	8

N-Vecinos con 40 Películas	precision	recall	f1-score	support
1 Excelente	0.000	0.000	0.000	2
2 Buena	0.250	1.000	0.400	2
3 Mala	0.000	0.000	0.000	2
4 Regular	0.000	0.000	0.000	2
avg / total	0.062	0.250	0.100	8

N-Vecinos con 30 Películas	precision	recall	f1-score	support
1 Excelente	0.000	0.000	0.000	2
2 Buena	0.250	1.000	0.400	2
3 Mala	0.000	0.000	0.000	2
4 Regular	0.000	0.000	0.000	2
avg / total	0.062	0.250	0.100	8

N-Vecinos con 20 Películas	precision	recall	f1-score	support
1 Excelente	0.000	0.000	0.000	2
2 Buena	0.250	1.000	0.400	2
3 Mala	0.000	0.000	0.000	2
4 Regular	0.000	0.000	0.000	2
avg / total	0.062	0.250	0.100	8

REFERENCES

[1] Rotten Tomatoes. (n.d.). Rotten Tomatoes: About. Retrieved June 2, 2020, from https://www.rottentomatoes.com/about

[2] IMDB. (n.d.). IMDb Top Rated Movies. Retrieved June 2, 2020, from https://www.imdb.com/chart/top/

[3] Bruno Stecanella. (2019, May 10). What is TF-IDF? Retrieved May 27, 2020, from https://monkeylearn.com/blog/what-is-tf-idf/

[4] DanJurafsky. (2018, October 28). Text Classification and Naïve Bayes . Retrieved May 20, 2020, from https://web.stanford.edu/class/cs124/lec/naivebayes.pdf

[5] Wikipedia contributors. (2020, May 11). Bag-of-words model - Wikipedia. Retrieved May 23, 2020, from https://en.wikipedia.org/wiki/Bag-of-words_model

[6] Onel Harrison. (2018, September 10). Machine Learning Basics with the K-Nearest Neighbors Algorithm. Retrieved May 18, 2020, from https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

[7] Peter Mills (2011). Efficient statistical classification of satellite measurements. International Journal of Remote Sensing.

[8] Predicting IMDB rating. (2016, October 3). Retrieved May 22, 2020, from https://www.kaggle.com/luisblanche/predicting-imdb-rating

[9] R. Dhir and A. Raj, "Movie Success Prediction using Machine Learning Algorithms and their Comparison," 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), Jalandhar, India, 2018, pp. 385-390, Retrieved May 18, 2020, from 10.1109/ICSCCC.2018.8703320

[10] Rui Mao, Hongzhi Wang, Xiaolan Xie, Zeguang Lu. (2019, October 31). Data Science. Retrieved May 29, 2020, from https://books.google.com.mx/books?id=X0mvDwAAQBAJ&dq=predictor+score+movie&source=gbs_navlinks_s

[11] Lobo, Jorge M., & Hortal, Joaquín (2003). Modelos predictivos: Un atajo para describir la distribución de la diversidad biológica. Ecosistemas, XII(1),1-8.Retrieved May 18, 2020, from https://www.redalyc.org/articulo.oa?id=540/54012119

[12] Mariñelarena-Dondena, Luciana, & Errecalde, Marcelo Luis, & Castro Solano, Alejandro (2017). Extracción de conocimiento con técnicas de minería de textos aplicadas a la psicología. Revista Argentina de Ciencias del Comportamiento, 9(2),65-76.Retrieved May 18, 2020, from https://www.redalyc.org/articulo.oa?id=3334/333452119006

[13] Salazar G., Erika J., & Ortega L., Oscar (2006). Sistema de Búsqueda Personalizada y Recomendación de Documentación Científica. Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial, 10(30),25-42.Retrieved May 18, 2020, from https://www.redalyc.org/articulo.oa?id=925/92503003

[14] Moreno Arboleda, Francisco Javier, & Duitama Muñoz, John Freddy, & Montoya Gómez, Luis Fernando (2016). Una propuesta para la clasificación emocional de un álbum a partir de la letra de sus canciones. Tecnura, 20(47),57-70.Retrieved May 18, 2020, from https://www.redalyc.org/articulo.oa?id=2570/257044050005

[15] Figuerola, Carlos G., & Zazo, Ángel F., & Rodriguez Vazquez de Aldana, Emilio, & Alonso Berrocal, José Luis (2004). La Recuperación de Información en español y la normalización de términos. Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial, 8(22),135-145.Retrieved May 18, 2020, from https://www.redalyc.org/articulo.oa?id=925/92582210