Escuela Técnica Superior de Ingenieros Informáticos

Universidad Politécnica de Madrid

Valoración de películas basada en Procesamiento del Lenguaje Natural y Deep Learning

Trabajo Fin de Máster

Máster en Inteligencia Artificial

AUTOR: Rubén Rodríguez Fernández
TUTORES: Emilio Serrano Fernández y Jacinto González Pachón

2017/2018

Lista de acrónimos

CNN Redes neuronales convolucionales

RNN Redes neuronales recurrentes

LSTM Long-short term memory

GRU Gated Recurrent Unit

IMDB Internet Movie Database

NTA Análisis de Texto basado en Redes

TMDB The Movie Database

MPAA Motion Picture Association of America

SVD Descomposición en valores singulares

ReLu Unidad linear rectificada

TF-IDF Frecuencia de término - frecuencia inversa de documento

TF Frecuencia del término

IDF Frecuencia inversa de documento

LDA Latent Dirichlet Allocation

LSA Indexación Semántica Latente

Dan2 Dynamic Architecture for Artificial Neural Networks

WoM Word-of-mouth

BN Batch normalization

t-sne T distributed stochastic neighbor embedding

ROI Retorno de la inversión

BPTT Backpropagation Through Time

ROCAUC Area bajo la curva de la Característica Operativa del Receptor

Resumen

La industria cinematográfica es un negocio multi-millonario, pero esta industria no está exenta de riesgos y sólo unas pocas películas consiguen ser rentables. Para reducir la incertidumbre sobre las inversiones, los productores e inversores se basan en la experiencia previa y en estudios de mercado, pero no es hasta el estreno de la película, cuando se descubre si ésta es rentable.

El presente Trabajo de Fin de Máster versa sobre la aplicación de modelos basados en *Deep Learning*, técnicas de Procesamiento del Lenguaje Natural y Análisis de Grafos con el objetivo de predecir la taquilla de una película. Las predicciones se realizarán en la etapa de Desarrollo, donde los productores deciden si producir una película, y la fase Pre-producción, donde se planifica la producción de la película, ya que en estas fases los recursos invertidos son reducidos y por tanto las predicciones pueden ser utilizadas por los inversores y productores para reducir la incertidumbre en la toma de decisiones.

Este problema se ha transformado en un problema de clasificación, discretizando la taquilla de forma binaria y multi-clase en 9 clases, de tal forma que todas las clases tengan el mismo número de películas. Los resultados obtenidos son: un *accuracy* de 74.4 % y 87.2 % en la clasificación binaria, y 32.9 % y 46.9 % en la multi-clase para Desarrollo y Pre-producción respectivamente. Mediante el uso de variables creadas con técnicas de Procesamiento del Lenguaje Natural y Análisis de Grafos, se han conseguido mejorar los resultados de Desarrollo y Pre-producción en un 13.7 % y 24.3 % respecto a los experimentos realizados utilizando únicamente las variables empleadas normalmente en la literatura.

La principal aportación del trabajo ha sido la creación de una representación densa de películas basadas en grafos, que tiene en cuenta interacciones entre variables categóricas con una gran cardinalidad como el género, estudio cinematográfico, escritor o secuelas de películas. Por otra parte, se han realizado aportaciones metodológicas, como la utilización de redes neuronales recurrentes para el procesamiento de sinopsis, y la utilización únicamente de variables calculadas de forma automática, en contraste con otros trabajos de Desarrollo que requerían de procesamiento manual. Por último, se ha tratado de reducir el sesgo introducido en otros trabajos mediante la utilización de variables agregadas a nivel de conjunto de datos, que pueden dar información de tendencias, o la partición del conjunto de datos en entrenamiento y prueba de forma aleatoria ignorando el componente temporal de las películas.

Summary

The film industry is a multi-million dollar business, but this industry is not risk-free and only a few films are profitable. Producers and investors rely on previous experience and market research to reduce the uncertainty about investments, But it is not until the film is released when the film is known whether it is profitable.

The present Final Master's Dissertation is about the application of Deep Learning models, Natural Language Processing techniques and Graph Analysis to predict film's box office. Predictions will be made at the Development stage, where producers decide whether to produce a film, and the Pre-production stage, where film production is planned, as the resources invested so far are reduced. Therefore, the predictions can be used to assist investors in the decision-making process.

This problem has been converted into a classification problem, by discretizing the box office in both binary and multi-class with 9 classes, so that all classes have the same number of films. The results obtained are: an accuracy of 74.4 % and 87.2 % in binary classification, and 32.9 % and 46.9 % in multi-class for Development and Pre-production respectively. By using the information extracted with Natural Language Processing and Graph Analysis techniques, the results in Development and Pre-production stages have been improved by 13.7 % and 24.3 % respectively, with respect to the experiments carried out using only the variables commonly used in literature.

The main contribution of the work is the creation of a dense representation of graph-based films, which takes into account interactions between categorical variables with a high cardinality such as genre, film studio, writer or sequels of films. On the other hand, we have made methodological contributions such as the use of recurrent neural networks for synopsis processing. Finally, we have tried to reduce the bias introduced in other works by using aggregate variables, which can give trend information, or by partitioning the data set in training and testing randomly ignoring the time component of the films.

Índice

1. Introducción y objetivos 1
1.1. Contexto 1
1.2. Objetivos 2
1.3. Trabajos relacionados 3
1.3.1. Actores y directores 4
1.3.2. Información textual 5
1.3.3. Presupuesto 6
1.3.4. Categorizaciones del contenido 7
1.3.5. Secuelas y adaptaciones 7
1.3.6. País de origen 7
1.3.7. Estudio cinematográfico 8
1.3.8. Datos de espectadores 8
1.3.9. Fecha de estreno 8
1.3.10. Efectos gráficos 9
1.3.11. Duración de la película 9
1.4. Contribuciones en el contexto de los trabajos relacionados 9
2. Antecedentes 11
2.1. Transformaciones 11
2.1.1. Variables categóricas 11
2.1.1.1. *One-hot encoding* 11
2.1.1.2. Codificación multi-etiqueta binaria 12
2.1.2. Variables textuales 12
2.1.2.1. Frecuencia de término - Frecuencia inversa de documento (TF-IDF) 12
2.1.2.2. *Word2vec* 13
2.1.3. Representación de grafos 15
2.1.3.1. *Node2vec* 15
2.1.4. Reducción de la dimensionalidad 16
2.1.4.1. SVD 16
2.2. Modelos 18
2.2.1. Redes Neuronales profundas 18
2.2.1.1. Entrenamiento 19
2.2.1.2. Funciones de activación 21
2.2.1.3. Técnicas de regularización 23
2.2.2. Redes neuronales recurrentes 24
2.2.3. Mecanismos de atención 26
3. Desarrollo 27
3.1. Obtención de datos 27
3.2. Pre-procesado de datos 28
3.2.1. Sinopsis 29
3.2.2. Presupuesto 31
3.2.3. Compañía de producción 31

3.2.4. Género 33
3.2.5. Fecha de estreno 34
3.2.6. Secuela 35
3.2.7. Escritor, Director y Actores 35
3.2.8. *Graph embeddings* 35
3.2.9. Clase a predecir: Ingresos de taquilla 37
3.3. Transformación de datos 38
3.4. Selección de variables 39
3.5. Minería de datos 40
3.5.1. *Word embedding* 41
3.5.2. *Spatial Dropout* 42
3.5.3. RNN 42
3.5.4. Agregación 42
3.5.5. Dense 43
3.6. Evaluación e interpretación de los resultados 43
4. Resultados y discusión 45
4.1. Etapa de Desarrollo 49
4.2. Pre-producción 51
5. Conclusiones y líneas futuras 55
A. Librerías utilizadas 63
B. Parámetros utilizados en la creación del Grafo 65

Índice de figuras

1. Arquitecturas cbow y skip-gram 14
2. Ejemplo de representación de *word embeddings* 15
3. Proceso de generación de *Node embeddings* 16
4. Ejemplo de descomposición SVD 17
5. Ejemplo de representación de una neuronal pre-alimentada con una capa oculta 19
6. Estructura de una red neuronal recurrente 25
7. Estructura GRU y LSTM 25
8. Histograma del presupuesto 31
9. Histograma del número de películas por estudio cinematográfico . . . 32
10. Diagrama de cajas de la taquilla por género 33
11. Histograma de la taquilla por meses 34
12. Arquitectura del modelo base 40
13. Arquitectura del modelo RNN 41
14. Arquitectura base + RNN 41
15. División del conjunto de datos 44
16. Error porcentual absoluto al estimar la taquilla agrupando las películas utilizando los umbrales de la clasificación multi-clase. 48

Índice de cuadros

1. Clasificación temporal de predictores 5
2. Ejemplo de representación *one-hot* 11
3. Ejemplo de representación binaria multi-etiqueta 12
4. Ejemplo de representación de documentos utilizando TF-IDF 13
5. Equivalencia de atributos entre el conjunto de datos descargado y el formato utilizado 29
6. Estadísticos descriptivos de la taquilla por estudio 32
7. Discretización binaria de la taquilla. 38
8. Discretización multi-clase de la taquilla 38
9. Resumen de las variables utilizadas 39
10. Resultados obtenidos en la clasificación binaria. 45
11. Resultados más importantes obtenidos en la clasificación multi-clase . 46
12. Descripción de los experimentos de de Desarrollo más importantes realizados 49
13. Descripción de los experimentos Pre-producción más importantes realizados 51
14. Parámetros utilizados para la creación del grafo de Desarrollo 65
15. Parámetros utilizados para la creación del grafo de Pre-producción v1 65
16. Parámetros utilizados para la creación del grafo de Pre-producción v2 65
17. Parámetros utilizados para la creación del grafo de Pre-producción v3 65

1. Introducción y objetivos

En esta sección se introducirá el contexto en el que surge el problema, y qué objetivos se plantean en el presente trabajo. Posteriormente, se explicarán los trabajos relacionados en la literatura, y cuáles son las aportaciones realizadas en el contexto de dichos trabajos.

1.1. Contexto

La industria cinematográfica de Estados Unidos generó en 2016 un total de 38.6 miles de millones de dólares, y se estima que ambas cifras crecerán en los próximos años [46]. Pero esta industria no está exenta de riesgos, y sólo unas pocas películas consiguen ser rentables [28]. Con el objetivo de minimizar estos riesgos, los productores realizan valoraciones económicas basadas en la experiencia e intuición, estudios de mercado, y análisis del desempeño de películas previas, pero estas valoraciones no están exentas de errores, y no es hasta el veredicto final, cuando la película se estrena, que se descubre si la película es rentable [19].

Numerosos académicos han tratado de realizar estas valoraciones de forma automática, pero se enfrentan a la complejidad de cuantificar cualidades abstractas como la creatividad y emociones, efectos gráficos y otro gran número de variables que definen el éxito o fracaso de una película, y que además, van cambiando a lo largo del tiempo. Estas valoraciones suelen ser menos precisas que aquellas obtenidas por expertos en el dominio, pero ese no es su objetivo, sino tratar de proporcionar una herramienta que trate de recibir la incertidumbre y ayudar en la toma de decisiones [40].

Las valoraciones se llevan a cabo a través de modelos predictivos, cuyo desempeño depende en mayor parte de los predictores. En el ámbito de la producción cinematográfica, estos predictores se podrían separar en 5 grupos, basándose en la clasificación de Steiff [59]:

- Desarrollo: En la fase de desarrollo el objetivo es obtener un guión, ya sea mediante la adquisición del mismo o a través de un proceso iterativo en el que una idea se va refinando hasta conseguir un guión. Esta fase involucra típicamente la evaluación de un gran número de guiones, de los cuales se termina seleccionando únicamente uno.

- Pre-producción: La fase de pre-producción involucra la planificación de la producción, como por ejemplo la elección de las localizaciones donde se rodará la película, o contratar a los actores y al equipo de producción.

- Producción: En la fase de producción se ruedan las diferentes escenas que conforman la película.

- Post-producción: Se montan las escenas rodadas en la fase de producción, editan y se añaden los efectos visuales. Al finalizar la fase de post-producción la película ya está terminada.

- Distribución: La distribución es la última etapa de la producción de una película, en la que se distribuye en los cines y otros medios.

1.2. Objetivos

Las valoraciones se pueden realizar siguiendo diferentes criterios, y dependerán del objetivo de las mismas. En el presente trabajo nos centraremos en el éxito de la película desde el punto de vista del inversor, es decir, el beneficio obtenido. La mayoría de los trabajos existentes se centran en predecir la taquilla discretizándola en grupos que usualmente van de *flop* a *blockbuster* [53, 57, 63, 64, 65]. En otros casos se trata como un problema de regresión, en el cual se suele aplicar una transformación logarítmica a la taquilla debido a que la mayoría de las películas se sitúan en torno a la media y hay muy pocas películas en los extremos [18, 33, 34].

Otra posible opción sería medir el éxito de la película utilizando el Retorno de la inversión (ROI), que mide la cantidad de dinero que hemos obtenido por cada unidad monetaria invertida. El problema de predecir el ROI, al igual que la taquilla, se ha afrontado tanto como clasificación [15, 37, 55] como regresión [19, 20, 37].

Los dos criterios de éxitos presentan problemas, por una parte la taquilla indica la cantidad de dinero obtenida por la película, pero eso no quiere decir que sea rentable ya que el presupuesto puede haber sido superior [37]. Por otra parte, el ROI indica lo rentable que es una película pero no cuanto dinero tienes que invertir para que siga siendo rentable, y es muy sensible a películas que han obtenido una taquilla media-alta con presupuestos pequeños.

En el presente trabajo se ha escogido la taquilla como variable a predecir, porque a pesar de necesitar el presupuesto para determinar si se trata de un éxito o fracaso, los inversores pueden saber cual es el dinero a invertir si quieren obtener un determinado beneficio, al contrario que con el ROI. Además, se simplificará el problema convirtiéndolo en clasificación a través de la discretización de la taquilla, lo que permitirá comparar los resultados con un mayor número de trabajos relacionados.

Como predictores de la taquilla se utilizarán las variables disponibles en las fases de Desarrollo y Pre-producción, ya que la inversión realizada hasta el momento es reducida, y por lo tanto, un buen modelo predictivo podría ayudar al inversor en la toma de decisiones. El uso de variables posteriores a la etapa de Pre-producción como Producción y Post-producción, tiene una utilidad reducida, debido a que los recursos invertidos en dichas fases es elevado.

1.3. Trabajos relacionados

Los trabajos relacionados con la valoración económica antes del estreno de la película pueden distinguirse en dos grupos dependiendo en la etapa temporal en la que se obtuvieron los predictores. El primer grupo está formado por predictores de Desarrollo, mientras que el segundo grupo contiene predictores Pre-producción y en Post-producción en menor medida. Los resultados de estos trabajos no son directamente comparables, ya que no existe un conjunto de datos estándar para probar los modelos, y cada trabajo utiliza muestras diferentes, de diferentes tamaños, años de producción o centrados en un país en concreto. A pesar de esta limitación, se seguirán las prácticas de anteriores trabajos donde es común comparar los diferentes resultados, asumiendo que siguen la misma distribución.

Los trabajos de Desarrollo se centran en determinar que película se va a producir entre un gran número de películas. En la actualidad, este proceso se realiza manualmente utilizando el desempeño de películas similares y la opinión de expertos, por lo que un sistema que consiga extraer información adicional puede ayudar a reducir la incertidumbre sobre el éxito de la película [19]. Los trabajos utilizando predictores de Desarrollo son los más reducidos, y se centran en el análisis de características textuales de guiones o *spoilers* [19, 20, 33]. Una limitación importante de éstos trabajos es que los datos utilizados provienen únicamente de películas que se han producido, que pueden tener algo diferente de aquellas que no se produjeron, y como consecuencia los resultados estarán inevitablemente sesgados [20].

Los resultados de Desarrollo más significativos fueron los obtenidos por Eliashberg et al. [20], en los cuales se diseñó un experimento de selección de portfolio y se comparó con métodos tradicionales. Los porfolios se crearon utilizando las películas con mayor ROI para cada uno de los modelos. El modelo que propusieron obtuvo un ROI del 134.1 %, mientras que los métodos tradicionales obtuvieron 83.4 %, lo que supone un incremento del ROI del 60 %. En el caso de Hunter et al. [33], los resultados estuvieron orientados a determinar la relevancia de la información extraída utilizando Análisis de Texto basado en Redes (NTA), la cual demostró un alto grado de significancia y un incremento en el coeficiencia del R^2 de entre un 6.2 % y un 9.4 % para las diferentes pruebas. La métrica R^2 mide la proporción de varianza sobre la media que el modelo es capaz de explicar.

La mayoría de los trabajos se centran en los predictores de la fase Pre-producción e incluyen también predictores Post-producción. Algunos de los predictores más utilizados son los siguientes: actores, presupuesto, clasificación Motion Picture Association of America (MPAA), precuelas y adaptaciones, géneros, fecha de estreno, duración o el número de cines en el que se va a estrenar la película [57, 63, 64].

En los trabajos con predictores Pre-producción y Post-producción se suele discretizar la taquilla en un número concreto de clases para convertir el problema en clasificación. La falta de una convención sobre el número de clases a utilizar y el

método de discretización dificulta la comparación entre trabajos de la literatura, por lo que obtener unos mejores resultados no es indicativo de un mejor modelo o predictores utilizados. Las métricas utilizadas normalmente en la literatura son *accuracy* y *1-away*, que es una variación del *accuracy* que incluye las clases adyacentes como acierto. Los resultados más relevantes son 75.2 % de *accuracy* de Ghiassi et al. [22], y 56 % y 90 % *1-away* de Delen and Sharda [16] para 9 clases, 76 % de *accuracy* de Zhang et al. [64] para 7 clases, y 68.1 % y 97.1 % de *1-away* de Zhang et al. [63] para 6 clases. Otros trabajos se han centrado en predecir el ROI, siendo el trabajo más relevante el realizado por Lash and Zhao [37], que realizó tanto experimentos de clasificación binaria obteniendo un 81.2 % de *accuracy*, como multi-clase con 3 clases obteniendo un 70 % y 72 % de *accuracy* para los diferentes pruebas.

Respecto a los modelos, se han utilizado una gran variedad, cómo regresión lineal [7, 40, 52], técnicas basadas en árboles de decisión [39, 64], redes neuronales [39, 53, 55, 57], LogitBoost [37], Dynamic Architecture for Artificial Neural Networks (Dan2) [22], o fusiones de varios métodos [16].

En los siguientes puntos se explicarán detalladamente los predictores más importantes de la literatura, mientras que en la Tabla 1 se aporta una visión condensada de dichos predictores y se indica a que etapa de la producción pertenecen. Además, se indican las variables que se utilizarán en el trabajo. Las variables no utilizadas se han añadido incluido en el análisis por completitud.

1.3.1. Actores y directores

El equipo de producción de una película, tanto el equipo técnico como el equipo artístico son los encargados de convertir un guión en una película, y por ello, tienen la responsabilidad de llevar la creatividad a la película [40]. Entre los principales integrantes del equipo de producción, se considera que los directores y actores son los que tienen una mayor influencia sobre el éxito de una película. Por este motivo, la mayoría de los trabajos han tratado de incluir información de trabajos previos de los directores y actores que pudiese servir de indicativo de actuaciones posteriores. Esta influencia podría cuantificarse de dos formas, por una parte, las preferencias del público por sus actores o directores favoritos, y por otra parte, se estaría incluyendo la decisión del actor/director de participar en la película.

Esta influencia es comúnmente denominada *star power*, y se ha tratado de cuantificar de diversas formas, que se podrían categorizar en 3 grupos: popularidad, beneficios y premios. La popularidad se ha medido principalmente a través de los *rankings* de usuarios de Internet Movie Database (IMDB) [3, 56, 63], de forma manual categorizando una película como sin estrellas, estrellas en ascenso/descenso o con estrellas [52], midiendo la popularidad como el número de resultados en Google para un determinado actor/director [63] o utilizando medidas de centralidad en un grafos de actores [61]. Además, también se ha reportado significancia en reportar el

Variable	Utilizada	Desarrollo	Pre-producción	Producción y posterior
Director y Actores	✓		✓	✓
Guionista	✓	✓	✓	✓
Información textual	✓	✓	✓	✓
Presupuesto	✓		✓	✓
Género	✓	✓	✓	✓
Clasificación MPAA	✓			✓
País de origen		✓	✓	✓
Secuela y adaptaciones	✓	✓	✓	✓
Estudio cinematográfico	✓	✓	✓	✓
Datos de espectadores				✓
Fecha de estreno	✓	✓*	✓	✓
Efectos gráficos				✓
Duración				✓

*Tab. 1: Clasificación temporal de los predictores más utilizados en la literatura en la fases en las que se encuentran disponibles. Las marcas indican que se encuentra disponible en la etapa temporal y en el caso de la columna "Incluida" que se va a utilizar en el desarrollo del trabajo. Los campos marcados con * se utilizan únicamente para el filtrado del conjunto de datos.*

star power de actores y actrices por separado [3]. Respecto a los beneficios, se han clasificado distinguiendo si pertenecen top-10 de actores/directores que mas beneficio han generado en los dos últimos años [40], clasificando las estrellas en alto valor, valor medio y valor bajo en base a los beneficios [50, 57], o el beneficio conjunto del director/actores en toda su carrera [53]. Por último, se ha utilizado información acerca de nominaciones [36, 40], o el porcentaje de directores/actores que han ganado un Oscar en el equipo [7].

1.3.2. Información textual

El guión es uno de los factores más importante, ya que contiene la trama de la película y se utiliza como discriminante para elegir entre distintas películas en la etapa de Desarrollo. La selección de guiones en la fase de Desarrollo se realiza manualmente, y a pesar de que algunos estudios han reportado la imposibilidad de predecir el éxito de una película, se ha demostrado que hay correlación entre el precio pagado por los guiones y el éxito financiero de la película, resumido en la famosa frase *"Somebody knows something"* [23]. Pese a esta importancia y los indicios de correlación, los guiones han sido raramente utilizados para valoraciones automáticas debido a la gran cantidad de texto, la dificultad de interpretar el texto y el reducido número de guiones disponibles al público.

Eliashberg et al. [20] fue el primero en analizar guiones, y extrajo dos niveles de información. El primer nivel tiene como objetivo extraer información del estilo y contenido del guión, y fue realizado utilizando un modelo de bolsa de palabras. Para reducir el efecto de las diferentes formas de una palabra, se utilizó *stemming*, que transforma las palabras a su forma raíz. Posteriormente, calcularon un índice de importancia de las palabras utilizando su frecuencia y aplicaron Indexación Semántica Latente (LSA) para reducir la dimensionalidad a dos componentes. El segundo nivel de información estaba compuesto por información semántica, que se centraba en la estructura del texto, y tenía atributos como el número total de escenas, número de diálogos o duración media de los diálogos.

Hunter et al. [33] tomó un enfoque diferente a Eliashberg et al. [20], y utilizo NTA. NTA, es un conjunto de técnicas que permite representar un texto como una red de conceptos de tal forma que es posible distinguir el significado de los mismos. Estas técnicas hacen dos asunciones; que el conocimiento se puede modelar como un conjunto de palabras relacionadas entre si, y que la posición de los conceptos en el texto proporciona información sobre el significado o tema del texto [32]. En este trabajo, utilizaron una versión semiautomática de NTA desarrollada por Hunter [32] que se basa en un análisis morfo-etimológico. Concretamente, se basa en la extracción de palabras en categorías conceptuales definidas por su etimología, y que se relatan entre si (en la red) basadas en su coocurrencia en palabras conocidas como compuestos multi-morfémicos. Posteriormente, cálculo el tamaño de la red, utilizando como definición de tamaño el número de nodos contenidos en el nodo principal.

Debido a la gran cantidad de texto, también se han tratado de utilizar sinopsis o *spoilers*, siendo este último un resumen extenso del guión de aproximadamente de 4 a 20 páginas, en el cuál se encuentran los detalles más importantes. Eliashberg et al. [19] utilizo un enfoque similar a su posterior trabajo Eliashberg et al. [20], pero utilizando Descomposición en valores singulares (SVD) para reducir la dimensionalidad a dos componentes. Por otra parte, Lash and Zhao [37] utilizó Latent Dirichlet Allocation (LDA), para extraer un conjunto de 30 tópicos.

1.3.3. Presupuesto

El presupuesto, junto al guión, es uno de los factores más relevantes para determinar el éxito de una película [3, 22, 40, 48, 50, 53, 55], ya que de él depende en mayor o menor medida el resto de variables, como los actores, equipo técnico, calidad y el dinero invertido en publicidad [40]. También es relevante, que en algunos casos el director es el encargado de buscar financiación o influye en el proceso, y por lo tanto, también podrán tener cierta correlación [40].

1.3.4. Categorizaciones del contenido

Las categorizaciones del contenido como MPAA o el género nos permiten tener una visión de como es una película, y se utilizan usualmente a la hora de escoger una película. La calificación MPAA, es una escala que establece la ideonidad por edades de un determinado público para la visualización de una película. Las películas aptas para todos los públicos pueden dar la impresión de que son muy infantiles, mientras que aquellas con un mínimo de edad están limitando el público [40]. La escala MPAA se utiliza como un atributo categórico [36, 40, 48, 57]. El género, representa la temática o temáticas principales de una película, y nos aporta información de su estilo, tono, ambientación o tema entre otros [36, 40, 52, 63].

También se han utilizado otro tipo de categorizaciones de contenido, como en el caso de [19, 20] donde se extrajo información del texto a través de cuestionarios. Los cuestionarios eran rellenados por un grupo de expertos a partir de los guiones o *spoilers*, y constan de preguntas como "El héroe de la historia tiene una clara motivación" o "El final es sorprendente e inesperado". Para evitar posibles sesgos, se utilizaron 3 expertos, y se transformaba el resultado en una variable numérica con un rango de 0 a 3 dependiendo del número de expertos que contestase afirmativamente.

1.3.5. Secuelas y adaptaciones

Otro aspecto de gran influencia es el rendimiento de las precuelas, o libros en el caso de las adaptaciones. Este tipo de películas son conocidas de antemano, y tienen ventaja en el pre-lanzamiento [40]. Un claro ejemplo de ventaja es la base de *fans* que tiene una saga o libro, que posiblemente vean la película independientemente de su calidad. La decisión de continuar una saga viene determinada principalmente por el rendimiento de las películas anteriores, y ha sido utilizado como predictor positivo en numerosos trabajos [40, 48, 50, 55]. Otros posibles indicadores son el número de películas en una saga o la distancia en años desde la película previa. Una saga con muchas películas puede que esté saturada y hacer que los espectadores pierdan el interés, por otra parte el número de años desde que salió la última película puede que disminuya este efecto [7].

1.3.6. País de origen

El país de origen también ha demostrado ser relevante en el éxito de una película, siendo Estados Unidos el país que produce las películas más rentables por un amplio margen. Las películas producidas en Estados Unidos suelen salir a mercados extranjeros, pero no es así en el caso contrario, donde muy pocas películas se estrenan en países distintos al de producción. Esto influye también en que la industria cinematográfica estadounidense esté más desarrollada, y el presupuesto que tienen disponible para las películas sea mayor. Además, dado que la mayoría de las películas que visualizamos son de origen estadounidense, los espectadores pueden tener un sesgo hacia dichas películas. Por estos motivos, algunos trabajos han tratado de

incluir información acerca del país de origen [22, 63], el idioma del país de origen categorizando en (estadounidense, no estadounidense y de habla inglesa o el resto) [62] o información acerca de la población del país y renta per cápita [48].

1.3.7. Estudio cinematográfico

Las películas son producidas por los estudios cinematográficos, que juegan un papel importante tanto en la producción como en la distribución de las películas. Los grandes estudios suelen tener acceso a mejores teatros y redes de distribución más extensas, por lo que es de esperar que tengan más beneficios [40]. El estudio cinematográfico se ha utilizado como variable categórica que determina si se trata de un estudio grande [40], o con mas granularidad comprobando si se trata de Buena Vista, Fox, Paramount, Sony, Universal, Warner Bross o el resto [17]. Las redes de distribución se han tratado de cuantificar utilizando el número de pantallas en las que se estreno la película [7, 22, 48, 57].

1.3.8. Datos de espectadores

El veredicto final sobre una película es otorgado por los espectadores, y por lo tanto es el predictor más importante justo antes y después del estreno. Las predicciones justo antes se pueden hacer utilizando la actividad y contenido en las Internet para medir el entusiasmo de los espectadores, mientras que las producciones a posteriori se suelen centrar en medir el grado de satisfacción (determinado normalmente sentimiento) en las *reviews* de los espectadores. La actividad en Internet se ha tratado de medir de varias formas, como el número de apariciones del nombre de la película en Google [63], el registro de actividad de la página de la película en Wikipedia [44], o la tasa de *retweets* en Twitter sobre el contenido de una película como fotos, *trailers* y otros materiales promocionales [4]. El sentimiento de los espectadores se ha medido analizando *reviews* de periódicos [36] o con las reviews y clasificaciones de plataformas como IMDB y Rotten Tomatoes [53, 55].

1.3.9. Fecha de estreno

La distribución asistencia no es uniforme a lo largo del año, y determinados meses como Mayo, Junio y Septiembre tienen una mayor afluencia [21]. Aunque la fecha de estreno no es un factor decisivo, afecta a la taquilla de las películas, a excepción de los *blockbuster* [40]. Los tipos de película tampoco son homogéneos a lo largo del año, y por ejemplo, en Mayo se suelen estrenar la mayoría de las superproducciones de *Hollywood*, y por lo tanto es uno de los meses en los que más taquilla se genera [21], por ello el mes se ha considerado un factor relevante [53, 63] así como la estación [22, 64]. Los festivos también influyen en la afluencia, ya que determinadas festividades como San Valentín tienen una mayor tasa de estrenos y audiencia de películas románticas, y en general tienen una audiencia mayor. Las festividades se han incluido como variable booleana indicando si la película se había estrenado en

un festivo [40, 63], o si la película se había estrenado en un mes con un festivo importante [55]. La audiencia tampoco es uniforme a lo largo de los días de la semana, y determinados días como viernes y sábado tienen una audiencia mayor, por lo que también se ha considerado el día de estreno [63].

Por otra parte, las películas tienen que competir con el resto de películas con fechas de estreno similares, especialmente aquellas películas con géneros similares, y un mayor número de películas influye negativamente en la taquilla [17]. La competencia se ha medido como la inversa del número de películas en un periodo de dos semanas respecto al lanzamiento [17, 55], con un periodo de una semana [63] o asociando un valor de competencia (alto, medio y bajo) a cada mes del año [50, 57].

1.3.10. Efectos gráficos

También se han utilizado aunque en menor medida datos sobre efectos especiales o la duración de la película. La cantidad de efectos especiales podría asociarse con el genero, ya que usualmente aparecen en películas de acción o ciencia ficción, pero ha demostrado tener un alto poder predictivo [57].

1.3.11. Duración de la película

La duración de la película puede darnos información de la complejidad de la trama o velocidad a la que se desarrolla, y además puede que influya en la decisión de ver una película [17]. La duración de la película ha sido utilizada como variable numérica [17, 22, 50].

1.4. Contribuciones en el contexto de los trabajos relacionados

Las aportaciones del trabajo son principalmente metodológicas, tanto la aplicación de modelos predictivos que no habían sido utilizado previamente como la utilización de técnicas novedosas para la extracción de información. Además, se tratará de reducir la información introducida al modelo que no se conoce en el momento de estrenar la película.

En la mayoría de los trabajos se está introduciendo información posterior al estreno de la película, lo que en teoría, hace que sus resultados mejoren. La mayor cantidad de sesgo es introducido en el particionamiento de conjunto de datos en entrenamiento y prueba, ya que esta usualmente de forma aleatoria. Los métodos de particionado aleatorios asumen que las observaciones en el conjunto de datos son independientes, pero en el caso de las películas esta asunción no es cierta, debido a que las películas tienen un componente temporal, y el desempeño de películas futuras aportar información sobre las actuales, por ejemplo, con géneros que se ponen de moda.

Otros ejemplos de este tipo de sesgo sería cualquier atributo que se calcule de forma agregada, como la competencia asociada a un determinado mes [50, 57] o semana [63] o para una película en general [22, 57], la asociación de una tasa de taquilla por género [63], o la relevancia de una estrella [22, 57, 64]. La introducción de valores agregados está aportando información de tendencia, por ejemplo, si una actriz/actor ha participa en 4 películas con una taquilla alta, no podemos decir que en la primera película que participa tiene una gran relevancia, ya que está aportando información de la taquilla que no es conocida.

En el apartado metodológico, desde nuestro conocimiento este es el primer trabajo que utiliza Redes neuronales recurrentes (RNN) para el procesamiento de la información textual referente a las películas. Los anteriores trabajos se centraron en representaciones basadas en tópicos [19, 20, 37], basadas en NTA [33] o extraer información mediante expertos [19, 20]. La extracción basada en tópicos extraen información similar al género [37], por lo tanto su utilidad se ve reducida si ya se utiliza el género de la película, mientras que los procesos de NTA [33] y la extracción de información mediante expertos [19, 20] tenía componentes manuales, lo cual limita su aplicabilidad. Mediante la utilizando de RNN, se puede extraer información que tiene en cuenta la distribución espacial, por ejemplo, películas con giros inesperados.

Por otra parte, se aplicarán técnicas de extracción de información en grafos novedosas como *node2vec*, que permite representar cada nodo del grafo con una representación numérica compacta, denominada *embeddings*, donde los nodos similares estarán situados cerca. Además, el procedimiento desarrollado para la creación del grafo, donde los nodos representan películas y las aristas se definen en base a variables categóricas y continuas como los guionistas, directores, actores o presupuesto. Esta representación, permite establecer interacciones complejas que no hubiesen tratándolas como variables categóricas debido a su elevada cardinalidad.

La extracción de características de grafos de películas ya se había utilizado previamente, extrayéndose información utilizando medidas de centralidad o estadísticos de los nodos [37, 61], seleccionados manualmente. Dicha selección, además de ser un proceso costoso, introduce un sesgo sobre los aspectos considerados más relevantes por el autor, y no además no cuenta con la expresividad de los *embeddings* donde las métricas de distancia tienen sentido.

Por último, los resultados obtenidos no mejoran respecto a los obtenidos previamente en la literatura, pudiendo ser la causa la limitación de utilizar atributos conocidos en Pre-producción, excluyendo información como la competencia o datos de distribución que han demostrado tener gran relevancia en la literatura. Por otra parte, la utilización de información obtenida mediante técnicas de Procesamiento de Lenguaje Natural y Análisis de Grafos, ha demostrado una mejora significativa respecto a los experimentos realizados que utilizan únicamente los atributos utilizados normalmente en la literatura.

2. Antecedentes

En esta sección se detallarán los algoritmos utilizados en el trabajo. En la primera sección se detallarán las transformaciones que se aplicarán a las variables con el fin de obtener una representación adecuada para los modelos predictivos. En la segunda sección se explicarán los modelos predictivos utilizados en el presente trabajo, así como técnicas para mejorar el rendimiento de los mismos.

2.1. Transformaciones

En esta sección se explicarán las transformaciones de variables utilizadas. Las transformaciones se han agrupado por la tipología de variable, siendo las dos primeras transformaciones para convertir variables categóricas y textuales en numéricas, y la tercera sección a técnicas para reducir la dimensionalidad de variables numéricas.

2.1.1. Variables categóricas

La mayoría de los modelos predictivos no soportan variables categóricas, y por lo tanto es necesaria transformarlas en variables numéricas en las cuales la noción de distancia y orden tengan sentido. A pesar de que las variables categóricas pueden ser representadas numéricamente de forma ordinal, este tipo de representación asume una jerarquía entre variables así como distancias entre categorías diferentes, que pueden no tener sentido dependiendo del problema. En los siguientes puntos se explicarán técnicas que permiten, dependiendo del tipo de variable categórica de entrada, convertir variables categóricas en numéricas sin imponer nociones de jerarquía ni distancias.

2.1.1.1. *One-hot encoding*

La representación *one-hot* permite representar numéricamente variables categóricas que toman una única categoría en cada instante, asumiendo que las categorías no tienen un orden y que son equidistantes. Formalmente, en *one-hot* se transforma una variable $\mathbf{x}$ con posibles valores $\{x_1, .., x_n\}$ en un vector de dimensión n, donde la posición i del vector tendrá valor 1 cuando la variable $\mathbf{x}$ toma el valor x_i y el resto será ceros.

Categoría	*One-hot encoding*		
	perro	gato	pájaro
perro	1	0	0
gato	0	1	0
pájaro	0	0	1
perro	1	0	0

Tab. 2: Ejemplo de representación one-hot.

2.1.1.2. Codificación multi-etiqueta binaria

La codificación multi-etiqueta binaria es similar a la codificación *one-hot*, pero soporta que una variable tome más de una categoría a la vez. Formalmente, en la codificación binaria multi-etiqueta se transforma una variable $\mathbf{x}$ que puede tomar uno o más valores $\{x_1, .., x_n\}$ en un vector de dimensión n, donde la posición i del vector tomará valor 1 cuando la variable $\mathbf{x}$ incluya la categoría x_i, el resto de elementos será 0.

Películas	Representación binaria multi-clase			
	Acción	Ciencia ficción	Horror	Comedia
Avatar	1	1	0	0
Get out!	0	0	1	1
Iron man	1	1	0	0
Zootopia	0	0	0	1

Tab. 3: Representación binaria multi-etiqueta para el género de una película.

2.1.2. Variables textuales

Las variables textuales al igual que las categóricas no pueden utilizarse directamente en la mayoría de los modelos predictivos, y es necesario transformarlas en una representación numérica previamente. Una posible opción sería asumir que las palabras son categorías, y aplicar las técnicas del apartado de transformaciones categóricas 2.1.1, pero debido al elevado número de palabras diferentes que tiene una idioma, la dimensionalidad del vector resultante sería muy elevada y haría que el modelo sufriese problemas de dimensionalidad. Por otra parte, estas técnicas asumirían que las palabras no guardan relación entre si, lo cual no es cierto, ya que por ejemplo la palabra 'coche' y 'moto', se parecen más que 'coche' y 'piedra'.

En las siguientes puntos se explicarán técnicas que permiten convertir documentos o palabras en representaciones densas, y evitan los problemas mencionados en el párrafo anterior.

2.1.2.1. Frecuencia de término - Frecuencia inversa de documento (TF-IDF)

Frecuencia de término - frecuencia inversa de documento (TF-IDF) es una técnica que permite expresar la relevancia de una palabra para un documento en un conjunto de documentos. TF-IDF está formado por la combinación de Frecuencia del término (TF), que premia la frecuencia con la que aparece una palabra en un documento, y Frecuencia inversa de documento (IDF) que penaliza aquellas palabras que son comunes en los documentos ya que no aportan menos información para distinguir entre documentos. Como resultado, TF-IDF asigna valores grandes a las palabras que sirven para discriminar un documento respecto a los otros.

$$tf\_idf(t, d, D) = tf(t, d) \times idf(t, D) \tag{1}$$

El término TF para un término t respecto a un documento d se suele calcular como el número de veces que un término aparece en un documento, $f_{t,d}$, dividido por el número de palabras totales en el documento, l_d. Existen diferentes variaciones de TF como utilizar únicamente el número de apariciones en un documento, ajustes logarítmicos o modificaciones que minimizan el sesgo en los grandes documentos.

$$tf(t, d) = \frac{f_{t,d}}{l_d} \tag{2}$$

El término IDF para un término t respecto a un conjunto de documentos D se calcula como el logaritmo del número de documentos, $|D|$, dividido por el número de documentos en los que aparece dicho término.

$$idf(t, D) = log(\frac{|D|}{1 + \sum_{d \in D} c(t, d)}) \tag{3}$$

Donde $c(t, d)$ es una función que toma valor 1 si el término t aparece en d y 0 en caso contrario.

Debido a la elevada dimensionalidad de los vectores producidos, TF-IDF se suele utilizar conjuntamente con técnicas de reducción de dimensionalidad como SVD, que se explicará en el apartado 2.1.4.1 y permiten identificar aquellos componentes que son mas relevantes para identificar los documentos.

En la Tabla 4 se muestra un ejemplo de TF-IDF utilizando los siguientes documentos d1 (t1 t1 t1 t2 t3 t4), d2 (t1 t2 t2 t3), d3 (t2 t3) y d4 (t5 t1).

Documento	Términos				
	t1	t2	t3	t4	t5
d1	0.817875	0.272625	0.272625	0.42712	0.000000
d2	0.408248	0.816497	0.408248	0.00000	0.000000
d3	0.000000	0.707107	0.707107	0.00000	0.000000
d4	0.538029	0.000000	0.000000	0.00000	0.842926

Tab. 4: Ejemplo de representación de documentos utilizando TF-IDF.

2.1.2.2. *Word2vec*

Word2vec es una herramienta que permite generar *word embedding*, que son una técnica de modelado de lenguaje que permite convertir una palabra en una representación numérica compacta donde tanto las operaciones aritméticas entre palabras como las distancias entre ellas son útiles [9]. Esto contrasta con técnicas utilizadas anteriormente como *one-hot encoding* que utiliza representaciones dispersas, y las

distancias entre palabras no son útiles debido a que asume que las palabras no guardan relación entre si.

En *word2vec* los *word embedding* se generan utilizando grandes conjuntos de textos y asumen que la estructura y significado de una palabra se puede determinar por su contexto, es decir, las palabras que rodean a dicha palabra. La representación se obtiene utilizando una red neuronal que tratan de predecir una palabra utilizando su contexto en el caso del modelo *cBoW*, o prediciendo el contexto utilizando una palabra en el caso de *skip-gram*. Concretamente, se maximiza la probabilidad en el conjunto de entrenamiento de que una determinada palabra w ocurra en un contexto c $P(w|c)$ en *cBoW*, o la probabilidad de que el contexto c ocurra dado una determinada palabra w $P(c|w)$ en *skip-gram*.

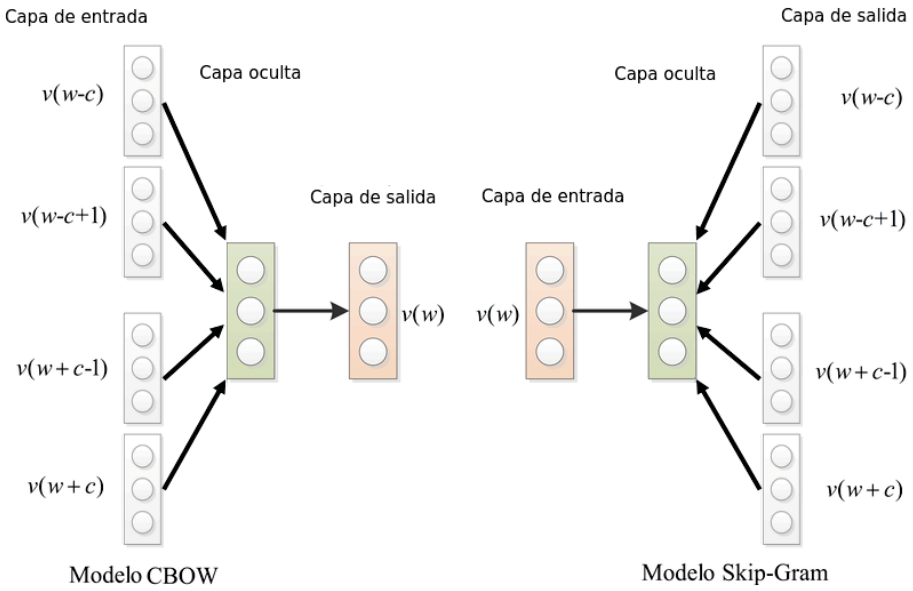

Fig. 1: Arquitecturas cbow y skip-gram. Imagen modificada a partir de [12].

Una vez entrenados los modelos tanto el modelo skip-gram como cbow, se elimina la última capa ya que es la capa oculta de la que se extraerá posteriormente la representación de las palabras. El tamaño de la representación es el número de neuronas en la capa oculta, que usualmente varía entre 50 y 300.

Como se puede apreciar en la Figura 3 a), las representaciones obtenidas tienen en cuenta el género, y se pueden realizar operaciones aritméticas entre las representaciones como $W(queen) = W(king) - W(man) + W(women)$, donde W representa el embedding de la palabra. La representación también es capaz de capturar relaciones gramaticales como se muestra en la Figura 3 b), así como capturar relaciones entre nombres Figura 3 c).

[1]Imagen tomada de: https://www.tensorflow.org/tutorials/word2vec

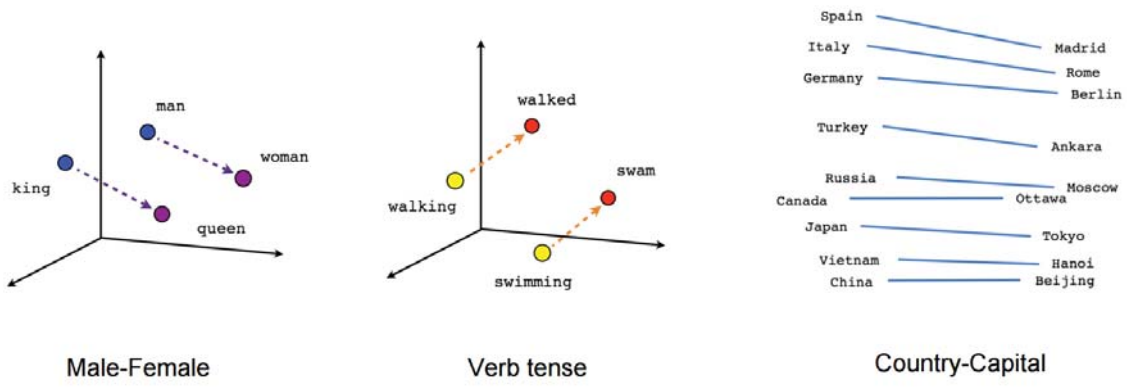

Fig. 2: Ejemplo de representación de word embeddings[1]*.Se ha aplicado la técnica de reducción de la dimensionalidad t-sne para reducir la dimensión de los vectores a 2 [42].*

El entrenamiento de estos modelos requiere de grandes cantidades de información y tiempo de entrenamiento, por lo que en la mayoría de las aplicaciones se utilizan tablas de equivalencias de palabras con su correspondiente embedding descargadas de páginas especializadas[2,3]. Estos modelos están entrenados usualmente con datos de dominio público, como la Wikipedia o Twitter, por lo que en determinadas aplicaciones que contienen terminología no muy común es posible que sea necesario entrenar de nuevo los modelos para introducir las nuevas palabras.

2.1.3. Representación de grafos

Las representaciones de grafos varían dependiendo del aspecto del grafo que se quiera enfatizar, como por ejemplo el grafo completo o los nodos individualmente. En el caso del grafo completo, existen representaciones numéricas inmediatas como la matriz de adyacencia, mientras que en los nodos individuales se utilizan estadísticos como el grado, pero su expresividad es limitada. Este apartado se centrará en las representaciones de nodos individuales, que se utilizarán en el presente trabajo, y se detallarán en los siguientes puntos.

2.1.3.1. *Node2vec*

La técnica *node2vec* desarrollada por Grover and Leskovec [24] es capaz de representar los nodos de un grafo utilizando una representación numérica compacta, y con propiedades similares a los *word embeddings*. El funcionamiento de *node2vec* se define en base a *word2vec*, primero se genera un corpus de nodos y después se introduce el corpus en *word2vec* obteniendo los *embeddings* de los nodos.

[2]https://nlp.stanford.edu/projects/glove
[3]https://code.google.com/archive/p/word2vec/

La construcción de un corpus se realiza mediante una estrategia de muestreo, donde para cada nodo se producen un número determinado de caminos aleatorios con una determinada longitud, obteniéndose una frase donde cada palabra es el identificador de un nodo. Dado que los grafos pueden ser cíclicos, se utilizan dos parámetros P y Q, que regulan la probabilidad de volver a visitar un nodo en el mismo camino aleatorio y la probabilidad de visitar nodos desconocidos respectivamente.

Una vez obtenido el corpus, que se compone de todos los caminos aleatorios generados, se introduce el corpus en *word2vec* y se obtiene un *embedding* para cada palabra, que se corresponderá con el identificador de un nodo.

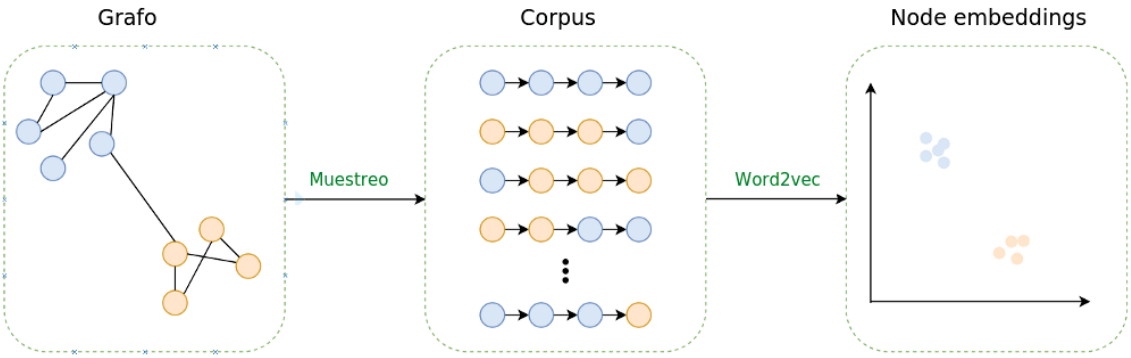

Fig. 3: Proceso de generación de node embeddings[4].

2.1.4. Reducción de la dimensionalidad

Las técnicas de reducción de dimensionalidad permiten proyectar los datos de entrada en un espacio con una dimensionalidad menor tratando de conservar la información mas importante. Estas técnicas son útiles en varios ámbito como la visualización de datos, ya que es difícil visualizar datos con más de tres dimensiones, o reducir la cantidad de variables a utilizar en un modelo predictivo. En el ámbito de los modelos predictivos, es útil para evitar tener variables correlacionadas, que en algunos modelos como regresión lineal puede ocasionar problemas de estabilidad, aumenta el rendimiento de los modelos al considerar menos variables, y en determinadas ocasionas reducir el sobre-ajuste. El siguiente punto se centrará en explicar una técnica de reducción de dimensionalidad que se utilizará en el presente trabajo.

2.1.4.1. SVD

SVD es una técnica de reducción de dimensionalidad basada en factorización de matrices. Concretamente, para una matriz $\mathbf{A} \in \mathbb{R}^{m,n}$ existe una factorización $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathbf{T}}$ tal que $\mathbf{U} \in \mathbb{R}^{m,r}$, $\mathbf{\Sigma} \in \mathbb{R}^{r,r}$, $\mathbf{V} \in \mathbb{R}^{n,r}$ es una matriz diagonal cuyos

[4]Imagen tomada de: https://towardsdatascience.com/node2vec-embeddings-for-graph-data-32a866340fef

términos se corresponden con los valores singulares de $\mathbf{A}$ y son estrictamente decrecientes y r es el rango de $\mathbf{A}$. Esta factorización siempre existe y es única, teniendo en cuenta la restricción a matrices con números reales.

Intuitivamente, utilizando como ejemplo una matriz en la que cada fila representa un documento y las columnas indican la presencia de un término en un documento, la matriz $\mathbf{U}$ (rotación) aprovecharía la similitudes entre documentos para obtener, por ejemplo, r conceptos/temáticas que representen los documentos y así reducir su número de columnas a r. La matriz $\mathbf{\Sigma}$ (escala) representaría la importancia de cada uno de los tópicos. Por último la matriz $\mathbf{V}$ representaría la transformación de términos a conceptos. En otras palabras, SVD trata de aproximar r vectores linealmente independientes utilizando una combinación lineal de k vectores. El número de componentes k se selecciona utilizando los k σ más grandes.

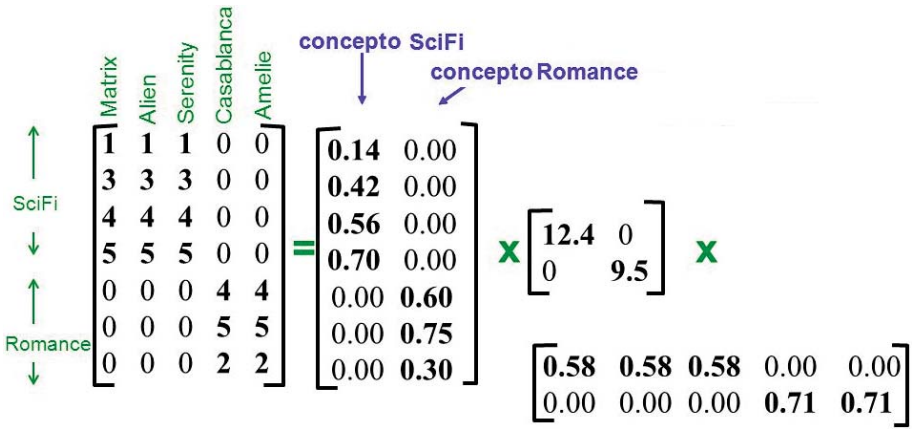

Fig. 4: Ejemplo de descomposición SVD[5].

La reducción de la dimensionalidad basada en SVD depende de los datos de entrada, y además, del algoritmo utilizado para obtener la factorización, que al ser aproximados suelen tener un componente aleatorio y puede sufrir de problemas de indeterminación de signos. Por lo tanto, continuando con nuestro ejemplo de documentos y términos, si utilizamos los k componentes extraídos por SVD para entrenar un modelo de clasificación, no podemos garantizar que si posteriormente utilizamos SVD para reducir la dimensionalidad de un nuevo conjunto de datos $\mathbf{B}$, los componentes extraídos y la importancia sean los mismos, y por lo tanto, el modelo de clasificación no funcionaría correctamente ya que la distribución de los datos sería diferente. Para solucionar este problema, se calcula el SVD de la matriz $\mathbf{A}$, obteniendo $\mathbf{U}$, $\mathbf{\Sigma}$ y $\mathbf{V}$, y la reducción de dimensionalidad a k componentes se calcula cómo $\mathbf{A}' = \mathbf{U\Sigma}$, mientras que los datos posteriores se obtendrían cómo $\mathbf{B}' = \mathbf{BV}$.

[5]https://slideplayer.com/slide/5283216/

2.2. Modelos

En esta sección se explicará primero el modelo predictivo utilizado en este trabajo, Redes Neuronales profundas, que se aplicarán en conjunción, con las variaciones de Redes Neuronales que se explicarán posteriormente. Los algoritmos de entrenamiento, funciones de activación y técnicas de regularización, que se explicarán en el apartado de Redes Neuronales profundas, serán aplicables a todas las variaciones sin ninguna modificación a menos que se especifique lo contrario.

2.2.1. Redes Neuronales profundas

Las redes neuronales artificiales es un algoritmo de aprendizaje automático que están bio-inspirados en la estructura del cerebro. La unidad básica en una red neuronal artificial es la neurona normalmente denominada unidad. Las neuronas artificiales están compuestas por un conjunto de entradas que se corresponden a las dendritas, una función de activación que determina si la neurona se excita ante una determinado conjunto de estímulos de las entradas, y las salidas que transmiten el estímulo que se corresponderían a los axones. Las neuronas tienen distintas afinidades a las diferentes dendritas, esto se refleja en las neuronas artificiales asociando un peso a cada entrada, que determinará la importancia de la misma para la neurona.

La entrada de una neurona se representa como un vector $\mathbf{x}$, correspondiendo el elemento el $\mathbf{x_i}$ con la entrada i. Cada entrada tiene un peso, representado como $\mathbf{w_i}$, con el cual se pondera la importancia de la entrada $\mathbf{x_i}$. Una vez ponderados los estímulos de entrada, estos se agregan sumándolos y se aplica una función de activación ϕ, que se explicarán en 2.2.1.2, para determinar el estímulo producido por la neurona. Matemáticamente, una neurona está representada por la siguiente expresión:

$$y = \phi(\sum_{i=0}^{n} w_i x_i) \tag{4}$$

Una gran limitación de los modelos con una sola neurona es la incapacidad de resolver problemas que no son linealmente separables, como el xor [45]. La introducción de redes neuronales multi-capa solventan este problema, siendo una capa un conjunto de neuronas que comparten las mismas entradas. Las redes neuronales multi-capa constan de una capa de entrada, que representa los datos de entrada, una o más capas ocultas, y la capa de salida, que proporciona la salida del modelo. En las redes neuronales pre-alimentadas, la salida de la capa t se convierte en la entrada de la capa *t+1*. A diferencia del perceptrón, las redes neuronales multi-capa con una sola capa oculta son capaces de aproximar cualquier función utilizando un número finito de neuronas bajo suposiciones leves en la función de activación, como por ejemplo, que la función de activación no sea lineal [31].

[6] https://en.wikibooks.org/wiki/Artificial_Neural_Networks/Feed-Forward_Networks

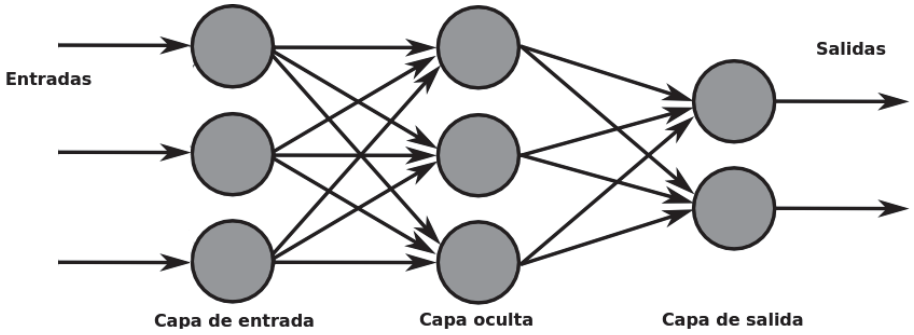

Fig. 5: Ejemplo de representación de una neuronal pre-alimentada con una capa oculta[6].

La construcción de un modelo basado en redes neuronales es un ciclo iterativo, primero se seleccionan los hiper-parámetros, como la arquitectura, funciones de activación, o las técnicas de regularización y parámetros que se verán en los apartados 2.2.1.3 y 2.2.1.1 respectivamente, después el modelo se entrena y se evalúa. El proceso se repite hasta que se alcanza un modelo satisfactorio, o se descarta.

La arquitectura es uno de los hiperparámetros más importantes de una red neuronal, en el caso de la red neuronal pre-alimentada, la arquitectura está constituida por el número capas y de neuronas en cada capa. Tradicionalmente el número de capas ocultas se limitaba a una y el número de neuronas era reducido debido al gran coste computacional y la gran cantidad de datos que necesita este tipo de modelos. En los últimos años, se ha producido una revolución debido al incremento de datos disponibles y sobre todo al incremento de la capacidad computacional gracias al uso de tarjetas gráficas y hardware especializado, volviendo viable redes neuronales artificiales con un gran número de neuronas y capas.

Las redes neuronales con más de una capa oculta son comúnmente denominadas redes neuronales profundas, y han ganado popularidad recientemente debido a su gran capacidad de generalización. La ventaja de utilizar redes neuronales con varias capas es que estas aprenden características jerárquicas, poniendo de ejemplo la detección facial, la primera capa aprendería estructuras básicas como bordes, la segunda capa aprendería formas básicas, la tercera capa formas como ojos y narices, y así irían aprendiendo poco a poco características cada vez más complejas hasta llegar a reconocer caras. En el caso de una red neuronal con una sola capa oculta, esta capa tendría que aprender la representación completa, y por lo tanto necesitaría un gran número de neuronas que incrementarían la propensión a memorizar el conjunto de datos, así como un incremento en el coste computacional.

2.2.1.1. Entrenamiento

El algoritmo de retro-propagación del gradiente es el algoritmo más utilizado en la actualidad para el entrenamiento de redes neuronales pre-alimentadas. A grandes rasgos, el algoritmo partiendo de la capa de salida, va propagando hacia atrás el

error de forma proporcional a la contribución que la neurona ha tenido en ese resultado. Los pasos del algoritmo de descenso del gradiente son los siguientes, y se ilustrarán, sin perder generalidad, con una red neuronal con una capa oculta.

1. El primer paso es introducir una observación en la red neuronal y propagarla hasta la capa de salida, donde obtendremos la predicción del modelo. Matemáticamente, en notación matricial sería:

$$net_1 = W_1 \times x + b_1; a_1 = f(net_1) \tag{5}$$

$$net_2 = W_2 \times a_1 + b_2; a_2 = f(net_2) \tag{6}$$

   Donde net es el estímulo total recibido por la neurona, W son los pesos de la neurona, b es el bias y f es la función de activación.

2. Una vez tenemos la predicción, se calcula el error para cada una de las capas empezando por la capa de salida

$$\delta^{(2)} = -(y - a_2) \circ f'(net_2) \tag{7}$$

   Donde f' es la derivada de f. El error en la capa de salida, se propaga a las capas anteriores a la capa anterior

$$\delta^{(1)} = (W_2^T \delta^{(2)}) \circ f'(net_1) \tag{8}$$

   Donde $\circ$ es el producto de Hadamark. En caso de tener más de una capa oculta, este último paso se repetiría propagando el error hacia atrás hasta llegar a la capa de entrada.

3. Una vez calculados los errores, se procede a calcular los gradientes con los cuales se ajustarán los pesos.

$$\nabla_{Wt} = \mu \delta^{(t+1)}(a_t^T) \tag{9}$$

   Donde t es la capa oculta para la cual se están calculando los gradientes y μ la tasa de aprendizaje.

La actualización de los pesos se lleva a cabo una vez se han acumulado los gradientes para un conjunto de observaciones del conjunto de entrenamiento. El número de observaciones utilizado se denomina tamaño de *batch*. En el algoritmo del descenso de gradiente, el tamaño del lote es igual al tamaño del conjunto de datos, por lo tanto se itera sobre todo el conjunto de datos antes de actualizar los pesos. Al utilizar conjuntos de datos grandes, el tiempo de convergencia es alto ya que a medida que crece el conjunto de datos más se tarda en actualizar los pesos.

En caso de utilizar un tamaño de *batch* igual a uno, se considera descenso de gradiente estocástico. El descenso de gradiente estocástico suele converger más rápido que el descenso de gradiente ya que se actualiza más frecuentemente. Se considera estocástico ya que se aproxima el gradiente total utilizando una única observación.

En un punto medio se encuentra el descenso de gradiente utilizando mini-lotes, en el que se utilizan número determinado de observaciones, normalmente entre 8 y 1024, para calcular los gradientes. Al igual que el descenso de gradiente estocástico, este método suele converger más rápido que el descenso de gradiente debido a que se actualizan los pesos más frecuentemente. Además, también suele ser más rápido que el descenso de gradiente estocástico gracias a la vectorización, que permite calcular el gradiente para un número reducido de observaciones simultáneamente sin costes adicionales excesivos. Al utilizar más observaciones para calcular el gradiente se consiguen mejores aproximaciones del gradiente, que usualmente compensan el costo computacional extra de incluir las observaciones adicionales.

La actualización de pesos se regula a través de la tasa de aprendizaje μ, que permite controlar la velocidad a la que estos cambian. Una tasa de aprendizaje baja hace que el algoritmo tarde más tiempo en converger o puede hacer que se quede atrapado en un mínimo local, mientras que una tasa de aprendizaje alta puede provocar que el modelo oscile en una función convexas. Para reducir el tiempo de convergencia y evitar las oscilaciones, se suele utilizar técnicas que modifican la tasa de aprendizaje a lo largo del entrenamiento. Un ejemplo de técnica de control de la tasa de aprendizaje la tasa de aprendizaje decadente, en la cuál se empieza con una tasa de aprendizaje alta que se va reduciendo a lo largo de las iteraciones, proporcionando exploración al principio del entrenamiento que se va reduciendo poco a poco para lograr explotación.

2.2.1.2. Funciones de activación

Las redes neuronales, al igual que el resto de modelo de aprendizaje automático, se utilizan para aproximar la función de densidad real f con una función g. En este contexto, el objetivo de la función de activación, también conocida como función de transferencia, es poder restringir la imagen de la función g. Dado que las redes neuronales se entrenan normalmente con técnicas que hacen uso de derivadas, la función de activación deberá ser diferenciable.

La función de activación más sencilla es la función identidad $f(x) = x$, y se suele utilizar en la última capa en problemas de regresión. El principal problema de la función de activación identidad, es que cualquier red neuronal pre-alimentada que utiliza únicamente la función de activación identidad es equivalente a una transformación afín, independientemente del número de capas utilizadas. Las implicaciones de esta equivalencia es la incapacidad de clasificar clases que no son linealmente separables, y que la arquitectura puede ser convertida en una red neuronal de una sola capa.

Debido a las limitaciones de las funciones lineales se utilizan funciones de activación no lineales, principalmente funciones sigmoidales, cómo la función sigmoidea o logística o la tangente hiperbólica. Las dos funciones anteriores toman valores en intervalos acotados [0, 1] ó [-1, 1] respectivamente, lo que las hace especialmente útiles en problemas de clasificación de dos clases. Además, debido a que las funciones aparecen en sus derivadas, estas son especialmente útiles debido a la disminución del costo computacional.

$$sigmoide(net) = \frac{1}{1 + e^{-net}} \tag{10}$$

$$tanh(net) = \frac{e^{net} - e^{-net}}{e^{net} + e^{-net}} \tag{11}$$

A medida que el número capas crece en las redes neuronales con función de activación sigmoidal se produce un fenómeno denominado desvanecimiento del gradiente, que consiste en el desvanecimiento del gradiente a medida que se propaga hacia atrás en la retro-propagación del gradiente. Como resultado sólo las capas finales se entrenan, ya que el error propagado hacia las primeras capas es cercano a cero. El desvanecimiento ocurre como consecuencia de que el error se propaga multiplicando los gradientes, como se aprecia en la expresión 8, y dado que la derivada de la función sigmoidal es siempre menor que uno, este valor se convierte en cero rápidamente.

La función de activación rectificadora Unidad linear rectificada (ReLu) reduce la probabilidad de desvanecimiento del gradiente y ha demostrado reducir el tiempo de entrenamiento [38]. La función de activación ReLu fue propuesta por Hahnloser and Seung [25] para redes neuronales dinámicas, y su efectividad en las redes neuronales profundas fue demostrada posteriormente por Nair and Hinton [47], siendo en la actualidad la función de activación más utilizada en las capas ocultas de las redes neuronales profundas. Otra de las ventajas de la función de activación ReLu es la dispersión, que ocurre cuando $net \leq 0$, y suelen obtener mejores resultados que las densas.

$$relu(net) = max(net, 0) \tag{12}$$

Durante el entrenamiento con funciones de activación ReLu, una actualización brusca de gradiente puede ocasionar que las neuronas acaben en un estado conocido como ReLu "muerta", en el cual la neurona no se volverá a activar. Una vez llegado a este estado, la neurona no se puede recuperar, ya que debido a que el gradiente si la neurona no se activa es cero, los pesos no se volverán a actualizar. Para solventar este problema se han introducido diferentes variaciones de ReLu como Leaky Relu [41] y PReLu [27], que permiten un pequeño gradiente en el caso de los valores negativos proporcional a net.

$$PReLu(net) = \text{máx}(0, net) - \alpha\, \text{máx}(0, -net) \tag{13}$$

Leaky ReLu es un caso particular de PReLu donde $\alpha = 0{,}01$. En el caso de PReLu, este parámetro se entrena con el resto de parámetros del modelo.

En problemas de clasificación donde el número de clases es superior a dos y las clases son mutuamente exclusivas se utiliza la función de activación *softmax* en la última capa para obtener la probabilidad de cada clase dada el patrón de entrada [11]. La función *softmax* es una generalización de la función sigmoidea y a diferencia del resto de funciones de activación opera sobre todo el vector de entrada. La probabilidad de la clase j dado el patrón de entrada $\mathbf{x}$ viene determinado por la siguiente ecuación:

$$P(y = c|\mathbf{x}) = \frac{e^{\mathbf{x}^T w_c}}{\sum_{k=1}^{K} e^{\mathbf{x}^T w_k}} \tag{14}$$

2.2.1.3. Técnicas de regularización

La selección de un arquitectura de una red neuronal es una de las mayores dificultades al solucionar un problema con redes neuronales, ya que se basa principalmente en la intuición y la prueba y error. Dado que a medida que aumenta el tamaño de una red neuronal también lo hace su expresividad, es común empezar probando con redes neuronales grandes de tal forma que el modelo sobreajuste los datos de entrenamiento. El sobreajuste en los datos de entrenamiento suele estar asociado a un error alto en el conjunto de datos de prueba, a partir de este punto se reduce el tamaño de la red o se aplican técnicas de regularización. Las técnicas de regularización tienen como objetivo hacer que el modelo generalice mejor, y por lo tanto, disminuya el error en el conjunto de datos de prueba.

Dropout es una técnica de regularización que desactiva neuronas aleatoriamente para forzar a la red neuronal a tener varias representaciones distribuidas [58]. En cada lote de datos, cada neurona se desactiva con una probabilidad determinada, convirtiéndose su salida en cero. Otra forma de interpretar el *dropout* es que las diferentes representaciones formadas forman clasificadores débiles, que después se promedian para producir un clasificador fuerte, similar a *ensembles* [29].

En casos donde las características tengan una alta correlación entre si, la eficacia de *dropout* se reduce ya que los valores reales de las neuronas convertidas en ceros se puede inferir con un alto grado de precisión. Este problema surge especialmente, en problemas como clasificación de imágenes o texto con *word embeddings*, donde los píxeles o los componentes del *word embedding* pueden guardar relación entre si. Para solventar esta limitación, se puede utilizar una versión de *Dropout* conocida como Spatial Dropout [60], que convierte elementos de la secuencia completa a cero. En el caso de la imagen, los elementos de la secuencia se corresponderían con los canales (rojo, verde y azul), mientras que en un texto se correspondería con una palabra.

Batch normalization (BN) es una técnica de regularización que normaliza la salida de las neuronas, y tiene como objetivo reducir el cambio de covariable [35]. Los parámetros de normalizado, media y desviación estándar, se van actualizando a la vez que se actualizan los pesos, pero dado que la actualización se hace de forma posterior se está introduciendo un pequeño ruido que produce efectos de regularización. El efecto del ruido disminuye a medida que aumenta el tamaño del lote, ya que las muestras son más representativas. Como resultado, los rangos numéricos se vuelven más estables, se reduce el tiempo de entrenamiento y se reduce la necesidad de otras técnicas de regularización.

2.2.2. Redes neuronales recurrentes

Las RNN son una variación de las redes neuronales que permite trabajar con datos secuenciales, especialmente útil en tareas como el procesamiento del lenguaje natural o la transcripción de audios. Este tipo de redes neuronales se denomina recurrentes debido a que se aplica la misma operación a cada elemento de la secuencia, teniendo en cuenta los elementos introducidos previamente. En los siguientes párrafos se explicará el funcionamiento interno de la versión básica de las RNN, conocidas como *Vanilla* RNN, sin perder la generalidad con variaciones más complejas de RNN como Long-short term memory (LSTM) y Gated Recurrent Unit (GRU).

Las RNN se alimentan de una secuencia $\{x_0, x_1, .., x_t\}$, que se procesa secuencialmente produciendo una secuencia $\{h_0, h_1, .., h_t\}$ y un conjunto de estados $\{s_0, s_2, .., s_t\}$. La secuencia de estados **s** representa la memoria de la neurona, y permite capturar relaciones con elementos previos de la secuencia. Esta memoria se calcula combinando el elemento de la secuencia **x** en el instante i y el estado **s** en el instante anterior i - *1*, por lo que la memoria se actualiza en cada instante de tiempo. En el instante $i = 0$ el valor del estado es 0. Las RNN cuentan con 3 parámetros, los primeros dos parámetros W_{ss} y W_{xs} permiten regular como se combinan s_i y x_i para producir el nuevo estado s_i. El tercer parámetro, W_{sh} transforma el estado en la salida h_i. En algunas implementaciones de *Vanilla* RNN, el parámetro W_{sh} es la identidad, por lo que $h = s$. Las ecuaciones que regulan la producción del estado y la salida en el instante i se muestran a continuación:

$$s_i = \sigma(W_{ss} \cdot s_{i-1} + W_{xs} \cdot x_i) \tag{15}$$

$$h_i = \sigma(W_{sh} \cdot s_i) \tag{16}$$

La función de activación se representa como σ, y se suele utilizar tanh o ReLu en el cálculo del nuevo estado regido por la expresión 15, mientras que para el cálculo de la salida neurona, expresión 16, depende del tipo de problema. Las funciones de activación utilizadas podrían ser cualquiera de las explicadas en el apartado 2.2.1.2.

En el entrenamiento de las RNN se utilizá Backpropagation Through Time (BPTT), que funciona desenrollando la RNN a lo largo de la secuencia de entrada como se puede apreciar en la Figura 6. Cada neurona recurrente en la secuencia

desenrollada se corresponde con una copia de la neurona recurrente original, que se comporta como una capa de una red neuronal tradicional. Una vez realizada esta transformación, la RNN se puede entrenar utilizando los algoritmos de entrenamiento explicados en la sección 2.2.1.1.

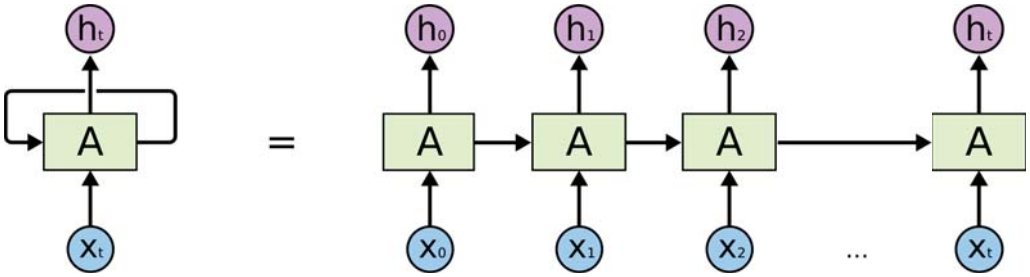

Fig. 6: Estructura de una neurona recurrente[7]*. A la izquierda se muestra su forma compacta, mientras que a la derecha se ha desenrollado con la secuencia x, que tiene t observaciones.*

En la actualidad, las *Vanilla* RNN no son muy utilizadas debido a una serie de problemas, como es la incapacidad de establecer relaciones temporales distantes, o problemas de desvanecimiento o explosiones de gradiente, acentuándose a medida que se incrementa la longitud de las secuencias [8].

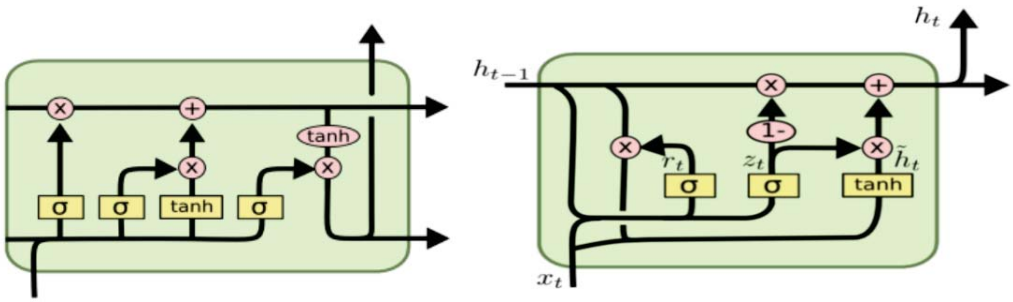

Fig. 7: Estructura de una celda LSTM y RNN respectivamente[8]*.*

Para solucionar dichos problemas, se desarrollaron variaciones más complejas de las RNN como LSTM [30] o GRU [13]. A alto nivel, tanto LSTM como GRU introducen una serie de operaciones, denominadas puertas, que controlan el flujo de información, y como efecto, solucionan el problema de las dependencias temporales distantes. En el caso de LSTM, se utilizan tres puertas, que intuitivamente controlan la información que se olvidará porque no se considera relevante (*forget gate*), que se

[7]Imagen tomada de https://magenta.tensorflow.org/2016/06/10/recurrent-neural-network-generation-tutorial

[8]Imagenes obtenidas de http://www.stratio.com/blog/deep-learning-3-recurrent-neural-networks-lstm/ y https://feature.engineering/difference-between-lstm-and-gru-for-rnns/ respectivamente.

aprenderá de la observación actual (*input gate*), y que información del estado actual se utilizará para calcular la salida (*output gate*).

Por otra parte, GRU no necesita una unidad de memoria como la LSTM y utiliza directamente el estado. A diferencia de las RNN, las GRU actualizan el estado de forma más inteligente asegurándose que mantiene información relevante previa, y en caso contrario se desecha.

No hay ningún criterio o escenario para la elección de GRU o LSTM, y se suelen probar ambos tipos de RNN. Por lo general, las GRU suelen obtener mejores resultados en conjuntos de datos pequeños, ya que el número de parámetros a estimar es menor, por otra parte, las LSTM suelen ser capaces de establecer relaciones más complejas en conjuntos de datos grandes.

2.2.3. Mecanismos de atención

Los mecanismos de atención permiten a las redes neuronales centrarse en las partes relevantes de los datos de entrada. Por ejemplo, en el caso de la traducción de texto utilizando RNN, los mecanismos de atención permitirían seleccionar las palabras que son relevantes para traducir una determinada palabra, sin tener que tener dicha información almacenada en el estado de la RNN.

La entrada del mecanismo de atención es un vector $\mathbf{y}$ que representa los datos de entradas. El contexto, representado por un vector $\mathbf{c}$, sirve para identificar las partes más relevantes de los datos de entrada en base a los elementos previos de la secuencia, y es un parámetro del modelo. Primero, se aplica el contexto c_i sobre el dato de entrada y_i obteniéndose la importancia del dato i m_i según la expresión 17. El resultado es un vector de importancias $\mathbf{m}$, que se escala de tal forma que todos sus elementos sumen 1 utilizando la activación *softmax* obteniéndose $\mathbf{s}$. Por último, se calcula la media ponderada z del vector de entrada $\mathbf{y}$ utilizando los pesos $\mathbf{s}$. El valor z representa la atención obtenida del vector $\mathbf{y}$ utilizando el contexto $\mathbf{c}$.

$$m_i = tanh(y_i \times c_i) \tag{17}$$

En caso de que la entrada al mecanismo de atención sea una matriz $\mathbf{Y}$ donde las columnas representan los elementos de la secuencia y las filas las características de la observación, el resultado de la atención sería un vector $\mathbf{z}$, donde el elemento z_i sería la atención para la característica en la fila $\mathbf{i}$ de la matriz $\mathbf{Y}$.

3. Desarrollo

En esta sección se explicará el desarrollo del trabajo que se ha realizado siguiendo el proceso KDD (Knowledge Discovery from Data), y contará con las siguientes secciones según [26]:

1. Obtención de datos: En esta fase se describirán las diferentes fuentes de datos utilizadas, así como los criterios utilizados en dicha selección.

2. Preprocesado del conjunto de datos: Los datos de obtenidos en la fase de obtención de datos se integran, y se realiza una limpieza y filtrado.

3. Transformación de datos: Los datos integrados se transforman en un formato adecuado para el modelo predictivo.

4. Selección de variables: Se seleccionan las variables que se utilizarán en los modelos predictivos.

5. Minería de datos: Se aplicarán diferentes modelos predictivos con el fin de resolver problema planteado en las fases anteriores.

6. Evaluación de los resultados: se explicarán las métricas con las cuales se evaluarán los modelos predictivos y la interpretación de las mismas.

3.1. Obtención de datos

Debido a la gran cantidad de conjuntos de datos disponibles sobre información de películas, la fase de obtención de datos se ha limitado a seleccionar un conjunto de datos. Los conjuntos de datos más usados, sobre todo en sistemas de recomendación de películas, son los proporcionados por la nueva interfaz de IMDB y *GrouLens*, pero estos carecen de información textual. Uno de los conjuntos de datos con mayor cantidad textual es el utilizado en el estudio de Bamman et al. [5], que utiliza las sinopsis de la wikipedia como información textual, pero ha sido descartado debido a que el tamaño de la sinopsis y calidad de la sinopsis tiene un claro sesgo con el éxito de la película. Por otra parte, IMDB proporciona uno de los mayores conjuntos de datos, tanto en número de películas como información, desgraciadamente, el conjunto de datos ha sido descontinuado por IMDB en favor del nuevo formato y dejará de estar disponible, por lo tanto afectaría la reproducibilidad del trabajo.

Dado que la mayoría de los conjuntos individualmente no proporcionan toda la información necesaria, se ha escogido un conjunto de datos que es una agregación de la información de *GrouLens* y The Movie Database (TMDB), que a pesar de tener un menor número de películas, contiene tanto información textual como información básica acerca de la película. El conjunto de datos está publicado en Kaggle [6] y consta de 45570 películas estrenadas antes de Julio de 2017. Los datos están divididos en 7 archivos:

- movies_metadata: Contiene la siguiente información básica sobre la película: saga a la que pertenece la película, un booleano que indica si se trata de una película de adultos, presupuesto, géneros, página web, id de la película, id de *imdb*, lengua original, título original, sinopsis, índice de popularidad, URL del póster, compañías productoras, países de producción, fecha de estreno, ingresos, duración, lenguas habladas en la película, título, media de votos y número de votos.
- credits: Los créditos contienen información del reparto, que está formado por los actores que actúan en la película o ponen voz a personajes, y el equipo, que está compuesto por el director, guionista, efectos especiales, entre otros. En cuanto al reparto, se incluye el género del actor y el orden de aparición en los créditos. En el caso del equipo, se incluye el género, y el departamento y posición al que pertenece.
- keywords: Son un conjunto de palabras clave que definen la película.
- links y links_small: Es una tabla de equivalencias entre el id de la película, y las referencias de TMDB y IMDB. La versión de links_small es un subconjunto de links.
- ratings y ratings_small: Contiene las valoraciones a nivel de usuario y la fecha en la que fue realizada. La versión de ratings_small es un subconjunto de ratings.

Otro de los aspectos decisivos al escoger el conjunto de datos es su libre distribución, lo que facilitará la tarea de replicar los resultados y permitirá compararlo con otros enfoques de forma fidedigna. El gran tamaño de la muestra permitirá la utilización de modelos que requieren gran cantidad de información como las redes neuronales profundas, y solventa uno de los principales problemas en los trabajos previos con predictores Pre-producción [19, 20, 33].

En el presente trabajo sólo se utilizarán los archivos *movie_metadata*, *credits* y *ratings*. Para facilitar la integración de nuevas formas de datos, los archivos de entrada se transformarán en un fichero *json*, que será independiente del origen de datos. Las adaptaciones están recogidas en la Tabla 5.

3.2. Pre-procesado de datos

Una vez obtenido el conjunto de datos, es necesario pre-procesarlo para conseguir una representación válida para el modelo predictivo a utilizar. La fase de pre-procesado se realizará conjuntamente con el análisis exploratorio, ya que las transformaciones a aplicar dependen de la distribución de los atributos. El análisis se realizará individualmente para cada característica producida, pudiendo estar compuesto de uno o más atributos del conjunto de datos. En algunos casos se propondrán distintas representaciones, que se valorarán posteriormente en el apartado de resultados.

Atributo	Origen
Presupuesto	movie_metadata.budget
Beneficio	movie_metadata.revenue
Sinopsis	movie_metadata.overview
Fecha de estreno	movie_metadata.release_date
Género	movie_metadata.genres
colección	movie_metadata.collection
Director	credits.crew
Actores y directores	credits.cast
Número de votos	ratings.numVotes
Puntuación media	ratings.averageRating

Tab. 5: Equivalencia de atributos entre el conjunto de datos descargado y el formato utilizado.

Respecto al filtrado del conjunto de datos, se han aplicado los siguientes filtros:

- Se filtraron aquellas películas de años anteriores al 2000 con el objetivo de minimizar el efecto de los cambios culturales.
- Se filtraron las películas que no tenían reportado el presupuesto o la taquilla.
- Para cada película, se cotejo la taquilla reportada con la disponible en BoxOfficeMojo y se filtro aquellas películas cuya taquilla se correspondía con la taquilla obtenida únicamente en los Estados Unidos.

El resultado del filtrado es un conjunto de datos con 2734 observaciones, que se utilizará en el resto del documento a no ser que se especifique lo contrario.

3.2.1. Sinopsis

La sinopsis es un pequeño resumen de la película, y suelen ser publicados por las distribuidoras para incitar a los espectadores a acudir al cine. La información contenida en la sinopsis es inferior a la presente en los guiones de las películas o en los *spoilers*, pero debido a la gran cantidad de información y el reducido número de guiones, es necesario un trabajo de pre-procesamiento muy complejo, que en ocasiones requiere de trabajo manual [19, 20, 33] limitando su aplicabilidad.

Las redes neuronales profundas no pueden procesar texto directamente, por lo que es necesario transformar el texto en una representación numérica. Para la sinopsis, se proponen dos representaciones, que se explicarán a continuación.

Las dos representación cuentan con una parte de pre-procesado conjunta, en la cual dependiendo de los parámetros utilizados, se llevarán a cabo las transformaciones:

- Filtrado de palabras vacías: se filtran las palabras vacías, también conocidas como *stop words*, ya que no suelen aportar demasiada información.
- Cambio a minúsculas: se convierten las palabras en letras minúsculas, para evitar diferencias entre palabras iguales pero con diferente capitalización.
- Filtrado de palabras no textuales: se borran aquellos *tokens*, entendiendo por *token* una secuencia contigua de caracteres delimitada por espacios, que no son palabras, como fechas o cantidades.
- Máximo número de palabras: se filtran aquellas palabras que aparecen con menos frecuencia en el texto.

La primera representación se obtendrá utilizando *TF-IDF*, y se aplicará posteriormente *SVD* para reducir la dimensionalidad y agrupar términos similares. La intuición detrás de esta representación es tratar de identificar un conjunto de tópicos (componentes) que representen la sinopsis. Los tópicos en éste contexto podrían entenderse como un conjunto de palabras que suelen estar asociados a un determinado tipo de película, como por ejemplo 'atraco', 'arma' y 'policía' a películas de acción. Por lo tanto, la información extraida debería ser similar al género de la película.

La primera representación no está teniendo en cuenta el orden de las palabras, lo que puede ocasionar que en algunos casos perdamos información valiosa, como por ejemplo negaciones o relaciones entre palabras. Es por ello, que en la segunda representación no eliminaremos la noción de secuencia, y simplemente convertiremos cada una de las palabras a un vector denso utilizando *word2vec*. El resultado es una secuencia de vectores, cada uno representando una palabra. Como se comentó en el apartado 2.1.2.2, la creación de *word embeddings* es un proceso muy costoso computacionalmente y requiere de una gran cantidad de datos, por lo que se utilizarán los *word embeddings* genéricos proporcionados por Google [9].

Los *word embeddings* proporcionados por Google están en formato binario, y se convirtieron a formato texto utilizando la librería gensim [54]. En el formato texto, en cada línea se encuentra la palabra y a continuación, separado por un espacio, la representación de la palabra. En caso de que una palabra no tenga representación, denominadas fuera de vocabulario, se le asigna un vector aleatorio, de la misma longitud que los *word embeddings*.

La longitud de las secuencias de *word embeddings*, al igual que la longitud de la sinopsis, variará de película a película. Dado que cada lote de datos contiene más de una secuencia de *word embeddings* (dependiendo del tamaño de *batch*), el lote generalmente no podrá ser representado como un tensor, que es un requisito indispensable para alimentar el modelo. Este problema se soluciona utilizando una técnica denominada *padding*, que consiste en rellenar con ceros todos las secuencias

[9] https://code.google.com/archive/p/word2vec/

hasta igualar la longitud de la secuencia más grande. En caso de que la longitud de la secuencia más grande sea muy superior al resto, o que simplemente se quiera reducir el tamaño de las secuencias para acelerar el entrenamiento las secuencias a veces se limitan a un determinado tamaño o reducir el uso de *padding*, en el presente trabajo se limitarán a una longitud de 200.

3.2.2. Presupuesto

Las películas con más taquilla suelen ser producidas por grandes estudios cinematográficos, debido a su mayor presupuesto, que influye en la calidad de la película, actores y actrices, directores y publicidad. El presupuesto se incluye en la mayoría de los trabajos Pre-producción y Post-producción debido a su relevancia.

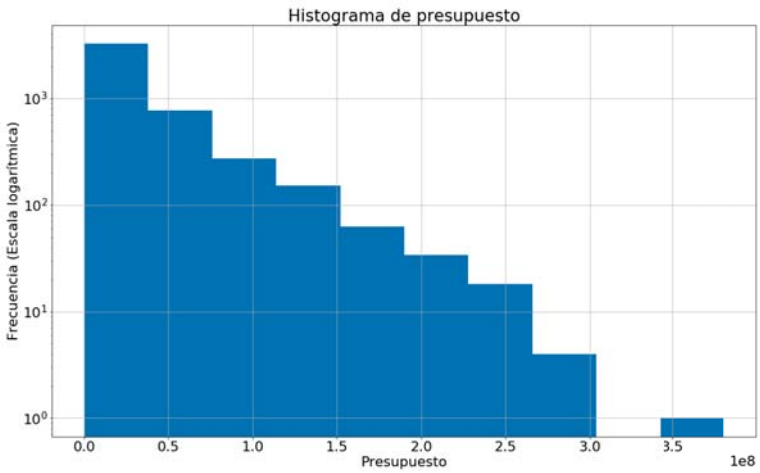

Fig. 8: Histograma del presupuesto.

Respecto al pre-procesado, la distribución del presupuesto se asemeja a una distribución exponencial, como se puede apreciar en la Figura 8. Las redes neuronales se comportan mejor con valores con distribuciones similares a una normal, además de reducirse el tiempo de entrenamiento. Para transformar la distribución del beneficio en otra que se asemeje a una normal se utilizará una transformación por cuantiles de la librería *sklearn*.

3.2.3. Compañía de producción

La compañía de producción es la responsable de transformar el guión en una película y juega un papel fundamental en su distribución y comercialización, por lo que se espera que tenga una gran influencia en el beneficio obtenido. En la Tabla 6 se muestran datos de los 12 estudios con taquillas medias más altas, y se aprecia como las taquillas medias varian considerablemente dependiendo del estudio.

Estudio cinematográfico	media	std	min	25 %	50 %	75 %	max
Marvel Studios	778.64	386.76	163.71	552.24	677.72	1008.36	1519.56
WingNut Films	646.16	405.78	29.36	291.38	871.37	957.21	1118.89
Pixar Animation	621.59	230.10	331.93	470.83	561.33	741.44	1066.97
Heyday Films	607.66	448.75	2.06	259.21	716.39	938.21	1342.00
Revolution Sun	579.70	380.17	105.32	268.52	557.18	766.96	1405.40
Dentsu	482.70	509.77	18.41	157.62	217.32	749.62	1513.53
Blue Sky Studios	479.46	234.90	246.23	282.78	408.58	580.56	886.69
Walt Disney Animation	465.62	371.13	14.46	219.79	377.35	643.03	1274.22
Bad Robot	439.82	564.32	1.43	95.91	301.78	521.11	2068.22
DC	438.20	390.85	1.69	59.64	343.08	764.35	1084.94
Temple Hill	429.36	269.07	46.43	305.91	348.32	704.16	829.00
Twentieth Century Fox	428.64	248.63	97.44	260.70	383.26	500.19	886.69

Tab. 6: Estadísticos descriptivos de los 12 estudios con la taquilla media más alta desde 1990. Todas las cifras, a excepción del el número de películas por estudio, se encuentran en millones de dolares.

La utilización del estudio cinematográfico como atributo plantea una serie de problemas, primero, el conjunto de datos no ofrece una jerarquía de compañías, y como consecuencia, *Marvel Studios* y *Marvel Enterprises* serán consideradas compañías diferentes, cuando pertenecen a la misma compañía. Por otra parte, como se puede apreciar en la Figura 9, la gran mayoría de películas son producidas por pequeños estudios, y el número de estudios cinematográficos es elevado, por lo que las codificaciones dispersas como *one-hot* no producirían buenos resultados.

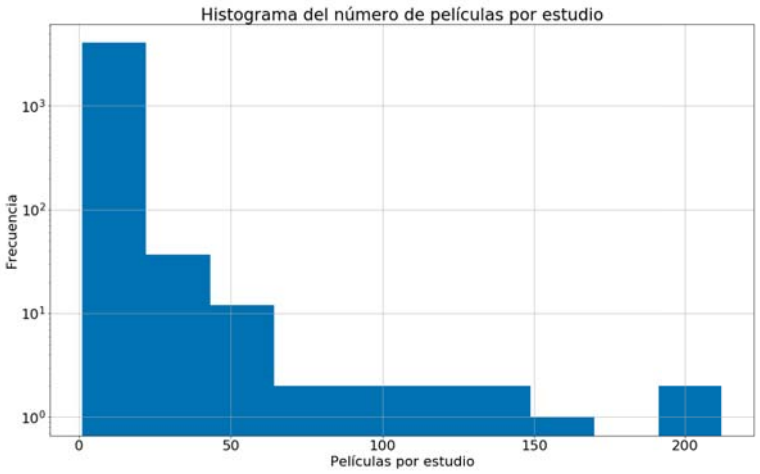

Fig. 9: Histograma del número de películas por estudio cinematográfico.

El problema de la dispersión se ha tratado en la literatura agrupando las com-

pañías de producción más importantes, y agrupando el resto en otra categoría [17, 40]. El problema de esta codificación es que no aporta información sobre los pequeños estudios, que son los que producen un mayor número de peliculas. Para solventar la falta de información de los estudios minoritarios, se incluirán como variables asociadas al estudio de producción el número de películas producidas y el beneficio medio hasta la fecha de la película. Esta codificación permite discriminar las grandes estudios debido a que tendrán un número de películas y beneficio medio superior, pero también aportará información sobre los estudios pequeños.

3.2.4. Género

El género de la película es uno de las variables más utilizadas en la literatura, tanto en estudios de Desarrollo, como Pre-producción y Post-producción. El género también es utilizado por los analistas de los estudios cinematográficos, que comparan las películas susceptibles de ser producidas con las películas más recientes de géneros similares [19].

La significancia encontrada en la literatura se puede verificar en la imagen 10, donde por ejemplo, las películas de animación y aventura son las mas taquilleras, mientras que las películas de drama y historia son las menos taquilleras.

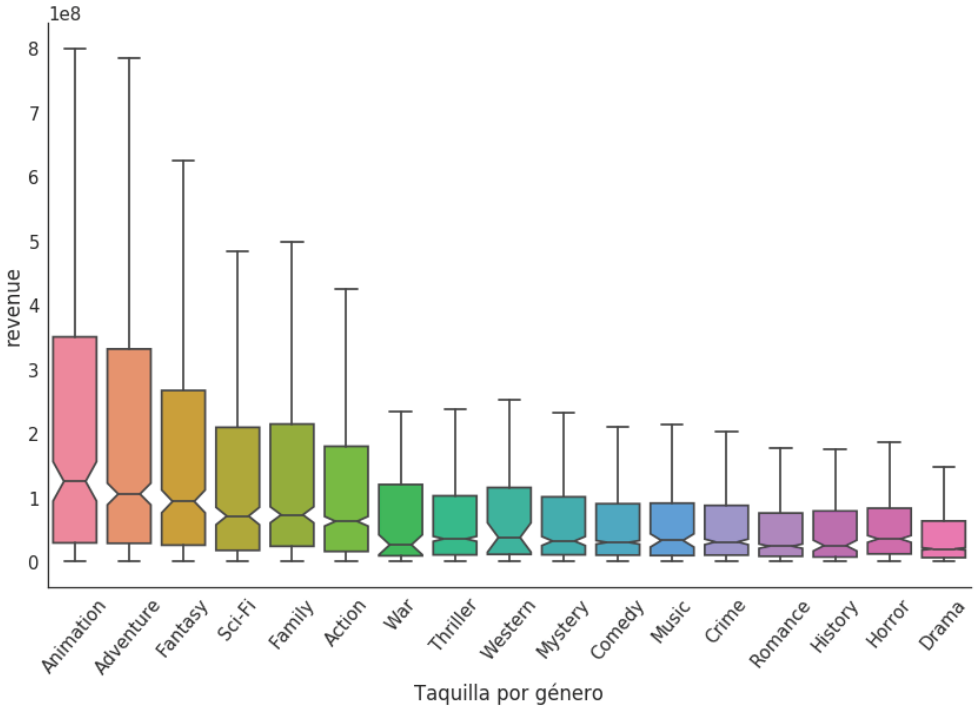

Fig. 10: Diagrama de cajas de la taquilla por género.

El género es un atributo categórico y necesita ser transformado en una representación numérica adecuada. Dado que en el conjunto de datos utilizado una película

puede pertenecer a más de un género, se utilizará una codificación binaria multi-etiqueta.

3.2.5. Fecha de estreno

La fecha de estreno es un factor importante para aquellas películas que no son super-producción, y ha demostrado una gran significancia en la literatura. En la gráfica 11 se puede observar como la distribución del beneficio a lo largo del año no es uniforme, y los meses de Mayo, Junio y Diciembre tienen una mayor taquilla. Este comportamiento había sido observado previamente por Follows [21], a excepción de Septiembre, que es considerado un mes de gran afluencia, sin embargo en el conjunto de datos utilizado es uno de los meses con menor taquilla.

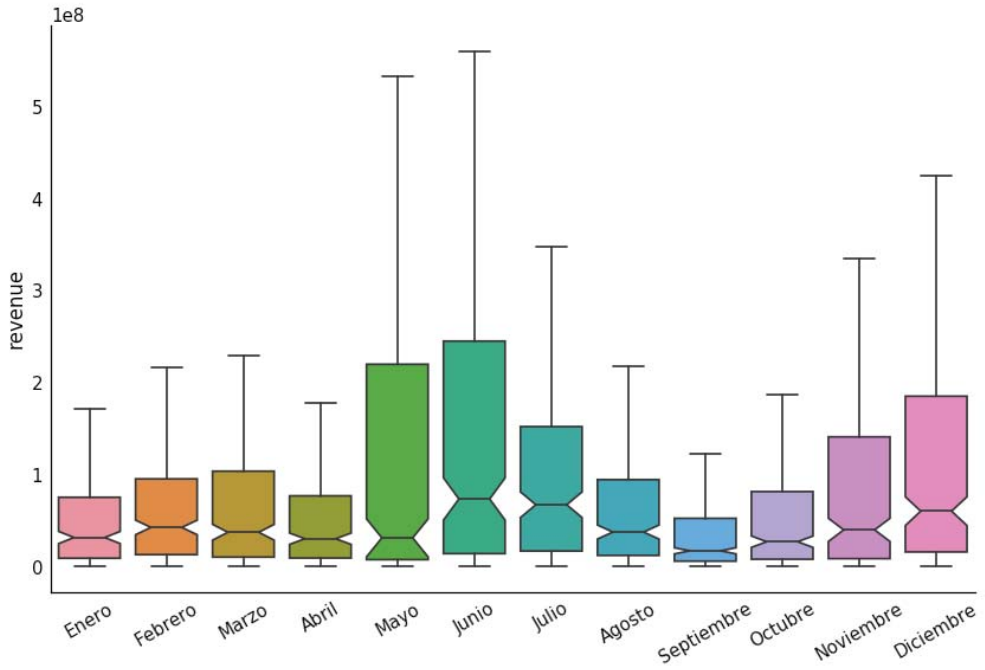

Fig. 11: Histograma de la taquilla por meses.

Por último, otro de los aspectos que más influencia tiene relacionado con la fecha de estreno es la competencia, pero debido a que el objetivo de este trabajo es realizar predicciones utilizando sólo datos previos a la producción no incluiremos este dato, ya que la fecha de publicación de las otras películas puede no ser conocido.

Para el pre-procesado de la fecha de estreno, se ha utilizado *pandas* para extraer el mes de estreno y posteriormente se ha convertido en una representación numérica utilizando *one-hot encoding*.

3.2.6. Secuela

Las películas que son secuelas parten con ventaja sobre a sus competidores, ya que son conocidas por el público [40]. Para el tratamiento de las secuelas se eligirá la opción mas utilizada en la literatura, y se convertirá en un factor que indica si la película es una secuela o no [40, 48, 50, 52, 55, 57].

3.2.7. Escritor, Director y Actores

Tanto los actores como los directores han sido ampliamente utilizados en la literatura y han demostrado tener una gran significancia, sin embargo, desde nuestro conocimiento, el escritor del guión no ha sido utilizado, a pesar de la gran relevancia que tiene el guión sobre el éxito de la película.

Al igual que las compañías de producción, las codificaciones dispersas no son de utilidad para los actores, directores y escritores, debido a la gran cardinalidad de dichas variables y el número reducido de datos disponibles. Por lo tanto, se escogerá una representación que agregue los datos de películas previas, concretamente, se utilizará la media del beneficio de las N últimas películas en las que participo el actor, director o guionista. En el caso de los actores, se reportará dicha media para los 3 actores más influyentes de la película por separado, mientras que para los directores y guionistas, en caso de haber más de uno, se reportará únicamente la mayor de las medias.

3.2.8. Graph embeddings

Las representaciones de películas basadas en grafos ya han sido utilizadas previamente en la literatura, pero sólo se han utilizado grafos que representan un aspecto a la vez, como grafos actores-película, actores-actores, en el cual dos actores tienen una arista si han participado en la misma película, y películas-películas, en la cual dos películas están conectadas si se han estrenado en fechas similares [61]. Por otra parte, también se han utilizado métricas, que si bien no se realizaron directamente sobre grafos, se podrían extrapolar, como la experiencia de un actor o grupo de actores en un género [37].

La representación basada en grafos propuesta, se centra en combinar varios aspectos a la vez e interacciones entre ellos. Concretamente, se creará un grafo cuyos nodos serán películas, y que se conectarán entre ellas en base a información sobre los actores, directores, escritores, género, compañía de producción y secuelas, que a partir de ahora denominaremos aspectos. Los aspectos a utilizar se podrán seleccionar, permitiendo crear grafos tanto de Desarrollo como Pre-producción. El grafo será dirigido, manteniendo la cronología de las películas, es decir, una película A sólo podrá estar conectada con otra película B, si A se ha estrenado después de B.

La importancia de las conexiones de las películas se modelizará mediante el peso

de la arista. Este peso, una vez transformados para que todos los pesos de las aristas de un nodo sumen 1, se utilizará en *node2vec* como la probabilidad de seleccionar una arista u otra en la creación del corpus.

La construcción del grafo consiste en dos pasos, primero se crea un grafo individual para cada tipo de aspecto, y después se agregan en un único grafo. La construcción de los grafos individuales se realizará a nivel de nodo (película), que se conectará con aquellas películas estrenadas previamente que maximicen la combinación de las métricas de relevancia. El número de películas máximo con el que se conectará una película dependerá del aspecto y está regulado por el parámetro *max_con*. Las métricas de relevancia se definen para pares de nodos o nodos individuales, siguiendo la siguiente nomenclatura: n es un nodo susceptible de ser conectado con el nodo actual *n_actual* en base a un aspecto. Se han utilizado las siguientes métricas de relevancia:

- importancia_temporal: Hace más relevantes aquellas películas cercanas en el tiempo. La relevancia temporal se calcula utilizando la expresión 18, y se escala en el intervalo (1 , 1.1).

$$importancia\_temporal(n\_actual, n) = diferencia\_años(n\_actual, n)^{-1} \quad (18)$$

- importancia_película: La importancia de una película n se ha definido como el número de votos de una película multiplicado por su puntuación media, siendo su distribución similar a la taquilla. Cabe destacar que sólo se calcula esta métrica para los datos de entrenamiento, ya que las películas del conjunto de datos de validación y prueba no se pueden conectar, por lo que no se está introduciendo información de la variable a predecir. Posteriormente, se le aplica un logaritmo a la importancia y se escala en un intervalo (1, 1.1).

- diferencia_presupuesto: La métrica de la diferencia de presupuesto se utiliza únicamente en grafos Pre-producción, y penaliza los nodos n que tienen una gran diferencia de presupuesto respecto al nodo *n_actual*. La métrica de diferencia de presupuesto se calcula siguiendo la expresión 19 donde $p(x)$ representa el presupuesto de x, y se escala posteriormente en un intervalo (1, 1.2).

$$diff\_presupuesto(n\_actual, n) = \log 1 + \frac{min(p(n\_actual), p(n))}{max(p(n\_actual), p(n))} \quad (19)$$

Intuitivamente, para un aspecto concreto las películas más cercanas son las más relevantes, pero estas pueden haber tenido una relevancia menor por algún motivo, por ejemplo, una pequeño estudio que lanza una película de superhéroes y consigue una taquilla baja, no es un indicativo de que la próxima película de *Marvel* vaya a obtener una taquilla baja. Para disminuir este efecto, se incluye la importancia de

la película, con la cual se consigue establecer relaciones más largas en el tiempo y favorece las interacciones con otros aspectos. Por último, para disminuir el efecto de la métrica de importancia de las películas se penaliza, en los grafos Pre-producción, aquellas películas con dispares, estableciéndose solo relaciones distantes cuando las películas tienen presupuestos similares.

La métricas de relevancia se combinan multiplicando las métricas individuales una vez escaladas. La contribución de cada métrica se regula mediante los parámetros de escalado, que se han optimizado manualmente mediante prueba y error, al igual que las expresiones utilizadas para medir los diferentes aspectos.

Una vez seleccionados los nodos $n = [n_1, n_2, ..., n_n]$ a los cuales se conectará n_actual, se calcularán los pesos w_i de cada una de las aristas que conectará n_actual con n_i utilizando la expresión 20, cumpliendo el requisito de que la suma de todos los pesos w_i sumen 1.

$$w_i = \frac{relevancia(n_{actual}, n_i)}{\sum_{j=0}^{n} relevancia(n_{actual}, n_j)} \tag{20}$$

La combinación de los grafos individuales se realizará agrupando las aristas de todos los grafos. La contribución de cada aspecto a un nodo está regulada por el parámetro *aspect_weight*, que será el valor máximo que puede alcanzar la suma de todos los pesos de las aristas de un aspecto dentro del grafo global. Si una arista está presente en más de un grafo, se agruparán y se sumarán sus pesos, enfatizando la relación entre las películas. Si un aspecto no tiene ninguna arista, como por ejemplo una película que no tiene precuelas, el *aspect_weight* de ese aspecto se distribuirá equitativamente a los otros aspectos.

Finalmente, el grafo creado se introduce en *node2vec* obteniendo un *embedding* para cada nodo.

3.2.9. Clase a predecir: Ingresos de taquilla

La taquilla de una película es el dinero obtenido en la venta de entrada de los cines, y representan la mayor parte del beneficio económico de una película. La taquilla no siempre es pública, y a veces se limita a determinadas regiones, como los Estados Unidos, por lo que es un factor limitante a la hora de incluir películas.

La taquilla será la variable a predecir, y al igual que en la mayoría de los trabajos en la literatura, se discretizará para convertir el problema en clasificación. Se proponen dos formas de discretizar la taquilla, la primera será binaria, que se utilizará para analizar la viabilidad del problema, mientras que la segunda será multi-clase, y siguiendo los trabajos de Delen and Sharda [16], Sharda and Delen [57] se discretizará en un total de 9 clases para facilitar su comparación con dichos trabajos.

En la discretización binaria, se define el punto de corte en la mediana, garantizando así que las dos clases tienen el mismo número de instancias. Los umbrales utilizados así como el número de instancias en cada clase se define en la Tabla 7.

Clase	Limite inferior (millones $)	Limite superior (millones $)	Instancias
+	-	54.7	1367
-	54.7	-	1367

Tab. 7: Discretización binaria de la taquilla.

Respecto a la clasificación multi-clase, la aplicación de los umbrales utilizados por Sharda and Delen [57] generan unas clases muy desbalanceadas, por lo que se ha optado por una discretización de tal forma que todas las clases tengan el mismo número de instancias. Los umbrales utilizados así como el número de instancias en cada clase se define en la Tabla 8.

Clase	Limite inferior (millones $)	Limite superior (millones $)	Instancias
C0	-	7.89	304
C1	7.89	16.2	304
C2	16.2	27.4	304
C3	27.4	42.4	303
C4	42.4	66.2	303
C5	66.2	96.9	304
C6	96.9	154.4	304
C7	154.4	277.7	304
C8	277.7	-	304

Tab. 8: Discretización multi-clase de la taquilla.

3.3. Transformación de datos

Las transformaciones presentada en la sección 3.2 se aplicarán al conjunto de datos explicado en en la sección 3.1 obteniéndose un conjunto atributos transformados. Dado que algunas de estas transformaciones, sobre todo las textuales y la construcción del grafo, son muy costosas computacionalmente y suponen un cuello de botella en el proceso de minería de datos, el proceso de transformación se realiza de forma previa al entrenamiento. El proceso de trasformación genera un archivo que contiene el conjunto de datos transformado y la variable a predecir, y que se utiliza posteriormente en el entrenamiento. La separación de ambos procesos también facilita la comparación de arquitecturas, ya que los datos utilizados serán siempre los mismos.

3.4. Selección de variables

En el presente trabajo no se realizará selección de variables debido a que el número de estas es muy reducido. Por otra parte, se realizarán pruebas tanto con variables de Desarrollo como Pre-producción, para tratar de cuantificar como disminuye la incertidumbre de la taquilla a medida que se acerca el estreno. Además, también se probarán diferentes combinaciones de variables dentro de ambos grupos, a fin de medir la relevancia de los predictores.

Para facilitar el reporte de los atributos utilizados, se ha creado una clasificación que depende de la tipología de las variables. Los atributos genéricos son aquellos que son generalmente utilizados en la literatura, mientras que los grupos grafo y texto se refieren al origen de los atributos. El resumen de las variables utilizadas y la fase y grupo al que pertenecen se muestra en la Tabla 9.

Atributo	Fase	Grupo	Descripción
summary_emb	D.	Text	Representación de la sinopsis utilizando word2vec
summary_tfidf	D.	Text	Representación de la sinopsis utilizando TF-IDF
graph_emb	D.-Pre.	Grafo	Representación del grafo de películas utilizando node2vec
presupuesto	Pre.	Gen.	Presupuesto de la película normalizado
género	D.	Gen.	Codificación multi-clase del género
avg_actor_1	Pre.	Gen.	Beneficio medio de las 3 últimas películas del 1º actor
avg_actor_2	Pre.	Gen.	Beneficio medio de las 3 últimas películas del 2º actor
avg_actor_3	Pre.	Gen.	Beneficio medio de las 3 últimas películas del 3º actor
avg_director	Pre.	Gen.	Beneficio medio de las 3 ultimas películas del director
avg_guionista	D.	Gen.	Beneficio medio de las 3 últimas películas del guionista
es_sequela	D.	Gen.	Indica si la película se trata de una secuela
mes	Pre.	Gen.	Codificación multi-clase del mes de estreno
num_estudio	D.	Gen.	Num. de películas producidas por el estudio
avg_estudio	D.	Gen.	Taquilla media del estudio.

Tab. 9: Resumen de las variables utilizadas. El tipo D. indica que se trata de una variable de Desarrollo, mientras que Pre. indica que se trata de Pre-producción.

3.5. Minería de datos

En este apartado se explicará como se aplicarán los modelos descritos en el apartado 2.2 a los datos transformados en el apartado 3.3. Se han propuesto 3 modelos, que se utilizarán en función de los datos de entrada, y se explicarán a continuación, mientras que las capas que forman los modelos se explicarán posteriormente.

La capa de salida en los 3 modelos propuestos es idéntica, y vendrá dada por el tipo de problema, es decir, dependerá del número de clases a predecir. En el caso de la clasificación binaria contará con 1 neurona, mientras que para la clasificación multi-clase con 9 clases contará con 9.

El primer modelo, denominado modelo base, se utilizará siempre que el conjunto de variables no incluya la variable *summary_emb*, que se denotarán como Datos entrada (1). El modelo es una red neuronal pre-alimentada con 1 o más capas ocultas (Dense), y una capa de salida (Dense). La arquitectura se muestra en la Figura 12.

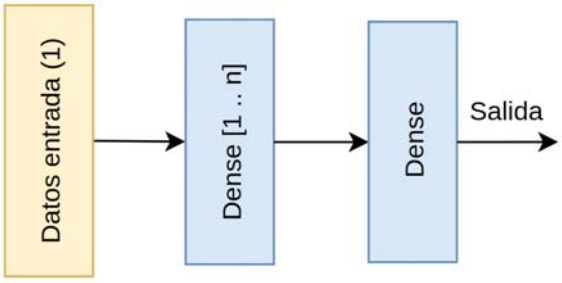

Fig. 12: Modelo utilizado cuando los atributos de entrada no incluyen summary_emb. Las entradas se denotan en color amarillo. La notación Dense [1..n] indica que puede haber de 1 a un número indeterminado de capas Dense.

Por otra parte, cuando los datos de entrada sean únicamente *summary_emb*, se utilizará un modelo basado en RNN, denominado modelo RNN, para procesar la secuencia de palabras. La entrada al modelo es una secuencia de índices, que se corresponden con palabras, y se transforman en una matriz de *word embeddings* donde cada columna representa una palabra utilizando la capa *Word embeddings*. A la matriz de *word embeddings* se le aplica la capa *Spatial Dropout*, donde algunas columnas se convertirán en ceros con una determinada probabilidad, y posteriormente se le aplicará una o maś capas *RNN*. La matriz procesada utilizando *RNN* necesita ser convertida en un vector para poder ser procesada por las capas *Dense*, por lo que se aplica la capa *Agregación* que convierte la matriz de entrada en un vector. Por último, se aplican una o mas capas *Dense* y la capa de salida. La arquitectura se muestra en la Figura 13.

Finalmente, cuando el modelo contiene tanto *summary_emb* como Datos entrada (1), se utiliza una variación de modelo RNN que integra los Datos entrada (1),

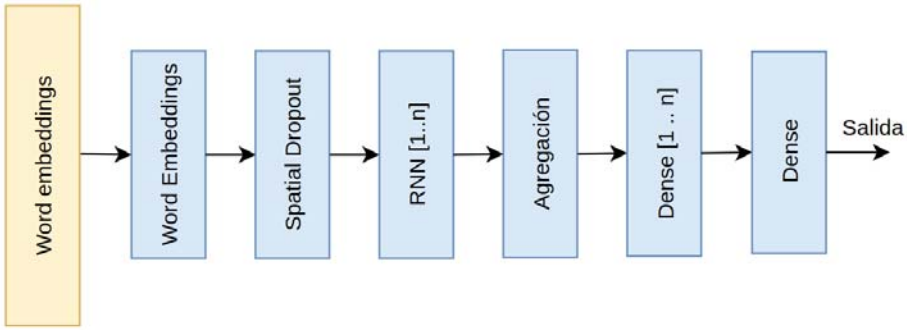

Fig. 13: Modelo utilizado cuando la entrada está formada por summary_emb. La entrada se denota en color amarillo. La notación Dense [1..n] y RNN [1..n] indica que puede haber de 1 a un número indeterminado de dichas capas.

denominado modelo base+RNN. Dicha integración se lleva a cabo concatenando la salida de la capa *Agregación* con los Datos entrada (1). La concatenación se define sobre el vector de salida de la capa de *Agregación* $\mathbf{a} = [a_1, a_2, ..., a_n]$ y el vector Datos entrada (1) $\mathbf{e} = [e_1, e_2, ..., e_m]$, siendo la concatenación de los vectores: $[a_1, a_2, ..., a_n, e_1, e_2, ..., e_m]$. Finalmente, se aplica al resultado de la concatenación una o más capas *Dense* y la capa de salida. La arquitectura se muestra en la Figura 14.

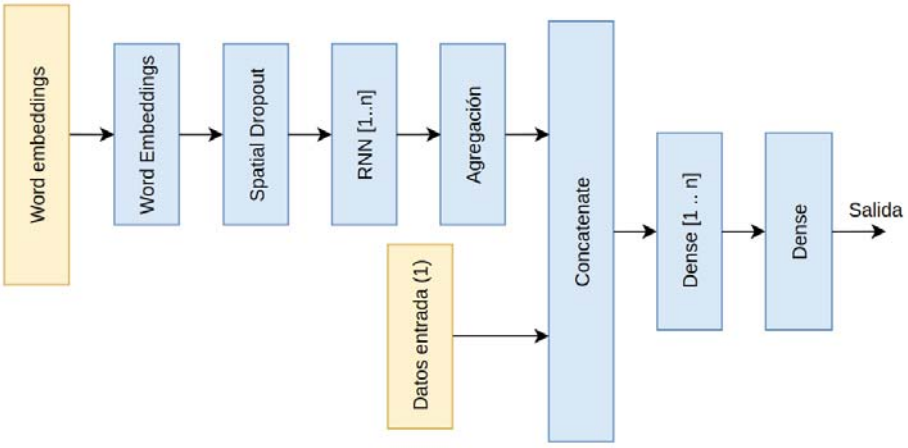

Fig. 14: Modelo utilizado cuando la entrada está formada por summary_emb y Datos entrada (1). La entrada se denota en color amarillo. La notación Dense [1..n] y RNN [1..n] indica que puede haber de 1 a un número indeterminado de dichas capas.

En los siguientes puntos se explicarán las opciones disponibles para cada una de las capas de los modelos explicados.

3.5.1. Word embedding

La capa de *word embedding* se implementa utilizando la capa *layers.Embedding* de Keras.

La entrada de la capa es una secuencia numérica, donde cada número de la secuencia indica la fila del *word embedding* en la matriz de *word embeddings*. La salida es una matriz, donde el número de filas es igual a la longitud del vector de entrada, y el número de columnas viene determinado por la dimensionalidad de los *word embeddings* utilizados.

3.5.2. Spatial Dropout

Para la implementación de la capa *Spatial Dropout* se utilizá *layers.SpatialDropout1D* de Keras. La capa se cuenta con un único parámetro, *rate*, determina la tasa de *dropout* a aplicar.

La entrada de la capa es una matriz, mientras que la salida es esa misma matriz donde algunas de sus columnas se han convertido en ceros, con probabilidad *rate*.

3.5.3. RNN

La capa *RNN*, cuenta con dos implementaciones diferentes. Las dos posibles opciones son *GRU* y *LSTM*, y sus implementaciones *layers.GRU* y *layers.LSTM* de Keras respectivamente. Los parámetros, comunes a las dos implementaciones, son los siguientes:

- *units*: Número de celdas recurrentes a utilizar en la capa. Si se proporciona una lista de tamaño n, se construirán n capas apiladas, donde el elemento i de la lista indicará el número de celdas recurrentes para la capa i.
- *dropout*: Tasa de *dropout* a aplicar después de cada capa de celdas recurrentes. En caso de no indicar ninguna tasa, no se aplicará *dropout*.
- *use_bn*: Indica si se debe usar *batch normalization* después de cada capa.
- *bidirectional*: Indica si las capas son bidireccionales.
- *activation_type*: Función de activación a utilizar, siguiendo la nomenclatura de las funciones de activación de Keras.

La entrada de la capa es una matriz, y la salida será una matriz con el mismo número de columnas y con *units* filas. Si *units* fuese una lista, el tamaño estaría representado por su último valor.

3.5.4. Agregación

La capa de agregación es la encargada de transformar una matriz en un vector y así poder aplicar la capa *dense* que representa la salida del modelo. La capa de agregación tiene 3 implementaciones disponibles, y el tipo de agregación se especifica utilizando el parámetro *type*.

La primera implementación es un mecanismo de atención, que se podrá utilizar con y sin contexto [1].

Los otros dos tipos de implementaciones son *global average pooling* y *global max pooling*, que calculan la media y máximo respectivamente, a nivel de fila. La implementación utilizada es *layers.GlobalAvgoling1D* y *layers.GlobalMaxPooling1D* de Keras.

La entrada de la capa será una matriz, mientras que la salida será un vector con dimensión igual al número de filas de la matriz de entrada.

3.5.5. Dense

La capa *Dense* representa una capa de una red neuronal pre-alimentada. La implementación utilizada es *layers.Dense* de Keras. La capa *Dense* tiene lo siguientes argumentos:

- *units*: Número de neuronas a utilizar en la capa. Si se proporciona una lista de tamaño n, se construirán n capas apiladas, donde el elemento i de la lista indicará el número de neuronas para la capa i.
- *dropout*: Tasa de *dropout* a aplicar después de cada capa de de neuronas. En caso de no indicar ninguna tasa, no se aplicará *dropout*.
- *activation_type*: Función de activación a utilizar, siguiendo la nomenclatura de las funciones de activación de Keras.

La entrada de la capa *Dense* es un vector, mientras que su salida es un vector de *units* elementos. Si *units* fuese una lista, el tamaño estaría representado por su último valor.

3.6. Evaluación e interpretación de los resultados

La minería de datos es un proceso iterativo, donde los modelos, parámetros y datos se van cambiando para tratar de mejorar el rendimiento de los modelos. Este proceso iterativo requiere de una forma de evaluar la bondad del modelo, que se realiza mediante métricas de evaluación. Las métricas de evaluación varían dependiendo del tipo de modelo, y del objetivo del mismo. En nuestro caso, nos centraremos en clasificación.

Las métricas de clasificación tratan de medir como de bueno es un modelo clasificando, y cada una de ellas enfatiza un aspecto diferente de la clasificación. Las métricas utilizadas son las siguientes:

- *accuracy*: Porcentaje de predicciones correctas.

- *precision*: Porcentaje de veces que el modelo habiendo predicho una clase acertó.
- *recall*: Porcentaje de predicciones correctas en una determinada clase sobre el número de veces que aparece dicha clase.
- *F1-Score*: Es una balance entre la *precision* y la *recall*.
- 1-Away: Es una extensión de la *accuracy*, en el cual se incluyen la clase inferior y superior como acierto. La métrica asume que existe un orden entre las clases.
- Area bajo la curva de la Característica Operativa del Receptor (ROCAUC): Mide la probabilidad de que las instancias se clasifiquen correctamente a medida que se cambia el umbral de discriminación.

Los umbrales utilizados para la división en clases están balanceados, por lo tanto, un modelo que imputase siempre la clase mayoritaria obtendría un *accuracy* de $1/número de clases$. En caso de obtener un *accuracy* superior, se podrá concluir que el modelo es capaz de encontrar cierta relación entre los datos de entrada y la clase a predecir.

Se dividirá el conjunto de datos en entrenamiento, validación, y prueba. El conjunto de entrenamiento se utilizará para entrenar el modelo, el de validación se utilizará para la selección de parámetros y arquitecturas basándose en las métricas vistas anteriormente, mientras que el conjunto de test se utilizará únicamente para reportar los resultados. En el apartado de resultados se reportarán los datos de test a no ser que se especifique lo contrario.

Como se comentó en el apartado de aportaciones, la elección de forma aleatoria de los conjuntos de entrenamiento, validación y prueba está aportando información de tendencia. Para minimizar este sesgo, se establecerá una división cronológica del conjunto de datos. La partición de entrenamiento se situará desde la primera película en orden cronológico y se extenderá hasta el 70 % de los datos. El 30 % restante, que se corresponde a validación y prueba a partes iguales, se seleccionará de forma aleatoria ya que no corremos el peligro de sesgo debido a que no se introducirán las variables objetivo en el modelo.

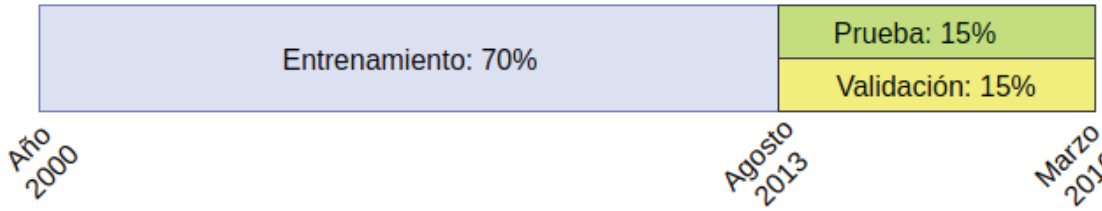

Fig. 15: División del conjunto de datos en entrenamiento, validación y prueba.

4. Resultados y discusión

En esta sección se detallarán los resultados obtenidos en las diferentes pruebas, y se tratará de dar justificar los mismos. Primero, se mostrarán los resultados obtenidos más importantes, y posteriormente, se mostrarán los resultados obtenidos tanto con información de Desarrollo como Pre-producción.

Los resultados se reportarán indicando si se tratan de datos de Desarrollo o Pre-producción, y se utilizarán todos los atributos disponibles en la fase especificada a menos que se especifique lo contrario. Para cada experimento se reportará la *accuracy* del modelo y *1-away* si se trata de clasificación multi-clase, y se calcularán para cada clase y de forma agregada la *precision*, *recall*, *f1-score* y *AUC*. Para la selección de modelos, se utilizará un reporte más compacto que constará de la *accuracy* y la métrica *1-away*.

En la Tabla 10 se reportan los mejores resultados obtenidos en los experimentos de clasificación binaria con datos de Desarrollo y Pre-producción. En ambos casos la *accuracy* es superior al 50 %, por lo que podemos concluir que el modelo es capaz de encontrar cierta relación entre los datos de entrada y la clase a predecir. Como era de esperar, a medida que aumenta la información disponible sobre la película, los resultados obtenidos por el modelo mejoran, obteniéndose una mejora de un 17 % al utilizar la información de la fase Pre-producción sobre la etapa de Desarrollo.

Datos	*accuracy*	Clase	*precision*	*recall*	*F1-Score*	*ROCAUC*
Pre-producción	87.2 %	-	0.87	0.86	0.87	0.94
		+	0.88	0.88	0.88	0.94
		media	0.87	0.87	0.87	0.94
Desarrollo	74.4 %	-	0.74	0.77	0.76	0.79
		+	0.75	0.72	0.73	0.79
		media	0.75	0.74	0.74	0.79

Tab. 10: Resultados obtenidos en la clasificación binaria.

A pesar de que la clasificación binaria no ha sido utilizada en la literatura debido a que aporta poca información sobre el éxito de la película, es útil para determinar en primera instancia si el problema es abordable. Una vez determinado que es abordable, se tratará en el resto de la sección la clasificación multi-clase, que ha sido estudiada en la literatura y nos permitirá comparar los resultados obtenidos.

En la Tabla 11 se muestran los mejores resultados obtenidos en la clasificación multi-clase. La diferencia de *accuracy* entre los modelos con datos de Desarrollo y Pre-producción se incrementa de un 17 % a un 42 %, confirmando que la fase de Pre-producción tiene una gran influencia sobre el éxito de la película. En el caso de la predicción multi-clase, el umbral de *accuracy* a superar sería 11.1 % debido a que

el número de clases sería 9. En ambos casos los modelos han alcanzado un *accuracy* superior a 11.1 %, por lo que concluimos que el modelo ha encontrado cierta relación entre los datos de entrada y la clase a predecir.

Datos	*accuracy*	1-away	Clase	*precision*	*recall*	*F1-Score*	*ROCAUC*
Pre-producción	46.9 %	80.6 %	C0	0.64	0.89	0.75	0.96
			C1	0.32	0.23	0.27	0.81
			C2	0.28	0.30	0.29	0.75
			C3	0.33	0.21	0.25	0.80
			C4	0.32	0.23	0.27	0.80
			C5	0.26	0.34	0.29	0.75
			C6	0.35	0.40	0.37	0.82
			C7	0.36	0.39	0.37	0.85
			C8	0.89	0.69	0.78	0.96
			media	0.47	0.47	0.46	0.86
Desarrollo	32.9 %	54.7 %	C0	0.55	0.69	0.61	0.90
			C1	0.20	0.10	0.14	0.73
			C2	0.12	0.10	0.11	0.67
			C3	0.08	0.02	0.04	0.62
			C4	0.16	0.13	0.14	0.66
			C5	0.11	0.09	0.10	0.64
			C6	0.19	0.13	0.15	0.71
			C7	0.20	0.17	0.18	0.71
			C8	0.41	0.82	0.55	0.90
			media	0.26	0.33	0.28	0.76

Tab. 11: Resultados más importantes obtenidos en la clasificación multi-clase

Un aspecto que destaca en ambos modelos es que las clases que están en los extremos, correspondientes con las películas de menores y mayores ingresos, tienen una mayor *precision*, mientras que en las clases centrales se obtienen peores resultados. Una posible interpretación de este resultado podría ser que tanto las películas exitosas como los fracasos se pueden determinar antes de su producción con un alto grado de *precision*, mientras que en el resto de las películas, su grado de éxito depende de factores que no se conocen durante el pre-estreno. Este comportamiento ya había sido observado previamente por Litman [40], en el cual los grandes éxitos no se veían afectados por factores como la competencia.

Teniendo en cuenta que las clases tienen un orden, la métrica *1-away* tiene especial relevancia, ya que nos indica si las predicciones en las cuales se ha equivocado el modelo se encuentran en las clases adyacentes o no. Por lo tanto, cuanto mayor sea la diferencia entre la métrica *1-away* y la *accuracy*, menor será la importancia de las instancias clasificadas incorrectamente. En la métrica *1-away* se obtiene un 80.6 % para los datos Pre-producción y 54.7 % en Desarrollo, lo que indica que el 63 % y el 32 % de las instancias mal clasificadas se encuentran en una clase adyacente

respectivamente, y por lo tanto, el coste del error es menor.

Los resultados obtenidos en este trabajo superan los reportados en Sharda and Delen [57], que utilizando datos Pre-producción y Post-producción obtuvo un *accuracy* de 36.7 % y *1-away* de 75.2 % obteniendo una mejora de un 27.7 % y un 7 % respectivamente. Por otra parte, los resultados no superan a los reportados en el trabajo de Delen and Sharda [16] con un *accuracy* de 56 % y *1-away* de 90.75 %, y Ghiassi et al. [22] con un *accuracy* de un 75.2 %. La diferencia entre los resultados podría deberse al uso de información Post-producción y de Distribución, como el número de pantallas en el cual se estrena la película, la competencia y duración, que han sido reportadas como variables importantes en sendos trabajos.

El principal problema a la hora de realizar comparaciones con los resultados de la literatura es la falta de un conjunto de datos estándar, por lo que obtener mejores resultados no es indicativo de un mejor modelo o variables utilizadas. El uso de conjuntos de datos pequeños, también relacionado con la falta de estandarización, dificulta también la comparación ya que a menor número de películas mayor será la probabilidad de que esta muestra no sea representativa. Por ejemplo, unos resultados mejores podrían deberse a que el modelo se adapta mejor a los datos seleccionados. Otro factor que acentúa el problema de los conjuntos de datos representativos es el uso de atributos que están disponibles para un número reducido de películas, como en el caso de Ghiassi et al. [22] con los gastos publicitarios previos al estreno, ya que el simple hecho de tener esa información puede estar aportándonos información sobre el desempeño de la película.

Dado el carácter continuo de la variable a predecir, una forma de proporcionar unos resultados más objetivos y que no dependan del proceso de discretización es tratar el problema como regresión. El modelo de regresión se creó utilizando la misma arquitectura y datos que el mejor modelo de clasificación, obteniéndose un error absoluto medio de 60.3 millones y un coeficiente R^2 de 0.79, que mide el porcentaje de varianza explicada por el modelo.

En la Figura 16 se muestra el error porcentual absoluto, que se calcula como el valor absoluto de la diferencia entre el valor predicho y real dividido por el valor real, obtenido al estimar la taquilla agrupando las películas mediante los umbrales de la clasificación multi-clase. La tendencia del error porcentual es decreciente a medida que las clases aumentan debido a que la diferencia entre el valor predicho y real aumenta. Para contextualizar este comportamiento, supongamos que tenemos una película cuya taquilla es 10 mil dólares y el modelo predice una taquilla de 20 mil dólares, obteniendo un error porcentual absoluto de 200 %, sin embargo, para obtener el mismo error porcentual absoluto en una película con una taquilla de 300 millones, el modelo debería predecir 600 millones. Como consecuencia, si el error porcentual disminuye a medida que las clases aumentan, también se reducirá el error absoluto.

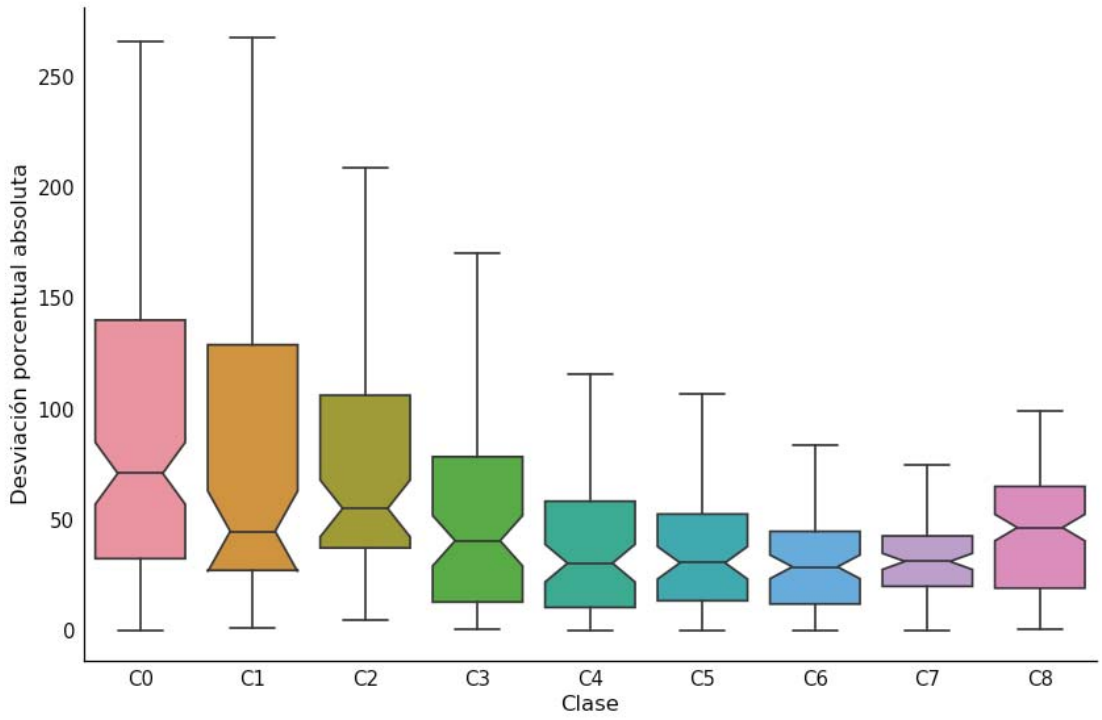

Fig. 16: Error porcentual absoluto al estimar la taquilla agrupando las películas utilizando los umbrales de la clasificación multi-clase.

Los resultados obtenidos en regresión son opuestos a los obtenidos en clasificación, ya que en este caso las clases con menor error porcentual son las centrales, mientras los extremos tienen un error porcentual mayor. En el caso del extremo izquierdo, se debe a que las cantidades son pequeñas, como se explico en el párrafo anterior. Sin embargo, en el extremo derecho este comportamiento indica que las predicciones dentro de esa clase no son muy precisas.

Analizando conjuntamente los resultados se pueden obtener principalmente dos conclusiones, por una parte los malos resultados de las clases centrales en clasificación a pesar de su menor error porcentual en regresión se deben a que las clases centrales tienen un rango más pequeño. Esto se debe a que el método de discretización utilizado divide en particiones con el mismo número de instancias sin tener en cuenta los rangos, y por lo tanto, las zonas con mayor concentración de películas tendrán rangos menores.

Por otra parte, el modelo es capaz de identificar en la mayoría de los casos cuando se trata de una película de la clase C8, pero no es capaz de dar una estimación precisa de la taquilla, y por lo tanto, si dividiésemos la clase C8 en sub-clases el error aumentaría considerablemente. Otro interpretación de los buenos resultados en la clase C8 sería que las taquillas más grandes tienen una dispersión muy grande,

y por lo tanto el rango de la clase C8 es superior al resto de clases.

Finalmente, debido a que los mayores errores se producen en la clase C8, el error absoluto obtenido es alto. Sin embargo, las películas en dicha clase sólo representan el 11.2 % y determinar la pertenencia a dicha clase sería suficiente en la mayoría de las instancias; por lo tanto, si filtramos las películas en la clase C8 el error absoluto obtenido será más representativo. El error absoluto excluyendo la clase C8 es de 22.7 millones de dólares, que se corresponde con el error del 88.8 % de las películas, y es considerablemente menor que el error obtenido en el conjunto de validación completo.

4.1. Etapa de Desarrollo

En la fase de Desarrollo los resultados obtenidos son de mayor utilidad que en el resto de las frases debido a que la inversión realizada en la mayoría de los casos es inexistente. Por otra parte, la información disponible en esta fase es reducida, y por lo tanto las predicciones suelen tener una mayor incertidumbre. Los experimentos más relevantes realizados se muestran en la Tabla 12.

Datos	*accuracy*	1-away	Información adicional
Todos (word2vec)	0.33	0.55	agg_type = average_pooling y grafo de Desarrollo
Todos (TF-IDF)	0.32	0.55	
Gen. + Text (word2vec)	0.30	0.53	agg_type = average_pooling
Gen. + Text (TF-IDF)	0.30	0.53	
Genéricos	0.29	0.54	
Grafos	0.25	0.52	grafo de Desarrollo
Text (Word2vec)	0.21	0.43	agg_type = average_pooling
Text (Word2vec)	0.20	0.41	agg_type = Atención con contexto
Text (Word2vec)	0.20	0.39	agg_type = Atención sin contexto
Text (Word2vec)	0.20	0.38	agg_type = max_pooling
Text (TF-IDF)	0.18	0.40	

Tab. 12: Descripción de los experimentos de Desarrollo más importantes realizados.

En los modelos cuya única variable de entrada son *word embeddings* se ha utilizado el modelo RNN con los siguientes parámetros: *Spatial dropout* ($rate = 0{,}2$), RNN ($units = 128$, $dropout = 0{,}2$), *Dense* ($units = [64, 32]$ y $dropout = 0{,}2$). Como función de activación se ha utilizado *ReLu*. El tipo de agregación utilizado se indica en cada experimento, mientras que el resto de parámetros como el uso de RNN bidireccionales, *batch normalization*, y más de una capa de *RNN* no mostró significancia.

Por otra parte, en los modelos cuya entrada son datos genéricos, grafos o ambos, se utilizo el modelo base con los siguientes parámetros: Dense ($units = [128, 64]$, $dropout = 0{,}4$) y función de activación *ReLu*. En los modelos con *word embeddings* que contenían datos genéricos, o datos genéricos y grafos, se utilizó el modelo base+RNN utilizando los parámetros reportados para los modelos con *word embeddings* cambiando el parámetro *Dense* ($units = [64, 32]$ y $dropout = 0{,}2$) por *Dense* ($units = [128, 64]$ y $dropout = 0{,}2$).

Los modelos se entrenaron durante un total de 400 *epochs* utilizando un tamaño de *batch* de 64, a excepción de los modelos que incluían *Word embeddings*, en los que se redujo el número de *epochs* a 100, ya que el costo computacional es superior y no mostraba signos de mejora a partir del *epoch* 80.

En la creación del grafo se utilizaron únicamente aspectos conocidos de Desarrollo, concretamente, se utilizó los escritores, género, compañía de producción y precuelas. La elección de los parámetros para la creación del grafo se realizó de forma manual, y se optimizó utilizando un enfoque de prueba y error. La lista de parámetros utilizadas se encuentra en el anexo B.

En los experimentos realizados, las variables textuales son las que peores resultados obtienen, estando comprendidos los resultados entre el 18 % y 21 % de *accuracy*. La diferencia entre los resultados utilizando TF-IDF y *word embeddings* es pequeña a pesar de la diferencia de expresividad de los datos utilizados, lo que podría indicar que el tipo de información extraída de los *word embeddings* es similar a la representación TF-IDF.

Los tipos de agregación utilizados no proporcionan una variación significativa en la *accuracy* del modelo, pero si en la métrica *1-away*, que mide la *accuracy* considerando las clases adyacentes como acierto. Por lo tanto, cuanto mayor sea el *1-away* mejor será el ajuste del modelo, y menor será el coste del error en clasificación. Un resultado destacable es que el tipo de agregación average_pooling funciona mejor que los mecanismos de atención, cuando los mecanismos de atención son una forma de realizar dicha media ponderando aquellos aspectos que son más importantes para la clasificación. Este resultado sugiere que la introducción de parámetros adicionales penaliza el rendimiento del modelo, y se debe a que el número de sinopsis utilizadas es pequeño. Por otra parte, el mecanismo de agregación max_pooling obtiene resultados similares al resto, lo que indica que las predicciones se basan en aspectos locales de la sinopsis.

Las variables textuales aportan información adicional que mejoran los resultados obtenidos con los modelos basados en grafos y generales, pero los resultados obtenidos con los *word embeddings* indican que están limitadas por el número de sinopsis utilizadas. La limitación del número de sinopsis es difícil de sobrepasar debido a que el número películas producidas es reducido. Esta limitación se ha tratado de

solventar de diferentes formas, como la utilización de varias sinopsis por película, pero en los experimentos realizados esto acentúa el sobre-ajuste. También se probó a aumentar el rango de películas incluidas, pero los resultados no mejoraron, pudiendo deberse a que las sinopsis previas a 1995 son mas breves y aportan menos información.

Los experimentos realizados con la información extraída de los grafos, a pesar de mejorar los resultados obtenidos por la información textual obtiene unos resultados peores que la información genérica. Dado que los grafos se crean exclusivamente con información genérica, los resultados deberían ser similares, y la diferencia en *accuracy* podría deberse a que el grafo no es capaz de capturar toda la información genérica. Otra posible interpretación, sería que debido al reducido número de información genérica de Desarrollo, las muestras seleccionadas para la construcción de aristas contienen algunas aristas entre películas que no son similares.

Finalmente, las combinaciones entre los diferentes grupos de datos mejoran los resultados, lo que indica que las grupos de características expresan diferentes aspectos relevantes para la clasificación de las películas. En el caso de los datos genéricos combinados con información textual, se obtienen los mismos resultados con TF-IDF y *word embeddings*, mientras que al incluir los datos del grafo el modelo que utiliza *word embeddings* obtiene una ligera mejoría.

4.2. Pre-producción

En la fase de Pre-producción la información disponible es mayor que en la fase de Desarrollo, por lo tanto se debería igualar o mejorar los resultados obtenidos. En esta fase, la información disponible es la información de la fase de Desarrollo más los actores, directores, posible fecha de estreno y presupuesto. Por otra parte, los resultados obtenidos en esta fase son menos influyentes, ya que a pesar de no haber comenzado a rodar la película, el estudio ha invertido recursos en la selección de actores, directores y planificación entre otros. Los resultados más relevantes obtenidos en la fase de Pre-producción se muestran en la Tabla 13.

Datos	*accuracy*	1-away	Información adicional
Todos (word2vec)	0.46	0.80	agg_type = average_pooling y grafo pre versión 3
Genéricos	0.37	0.65	
Grafo	0.36	0.67	Grafo pre versión 3
Grafo	0.35	0.61	Grafo pre versión 2
Grafo	0.33	0.62	Grafo pre versión 1

Tab. 13: Descripción de los experimentos Pre-producción más importantes realizados.

En los modelos con datos genéricos, grafos o ambos, se utilizó un modelo similar al apartado de Desarrollo, pero el número de neuronas de la primera capa *Dense* se ha incrementado, ya que debido a la mayor cantidad de datos, el modelo tiene que capturar relaciones más complejas. El parámetro utilizado es *Dense* ($units = [256, 64]$ y $dropout = 0{,}4$).

En los modelos con *word embeddings* que contenían datos genéricos, o datos genéricos y grafos, se utilizó el modelo base+RNN utilizando los parámetros reportados en la fase de Desarrollo, ya que la información textual es la misma, y se cambio el parámetro *Dense* ($units = [64, 32]$ y $dropout = 0{,}2$) por *Dense* ($units = [256, 64]$ y $dropout = 0{,}2$).

Se utilizó el mismo tamaño de *batch* y *epoch* que en la fase de Desarrollo, ya que se reportaron comportamientos similares en el entrenamiento.

En la creación del grafo, a parte de la información utilizada en la fase de Desarrollo, se utilizó los actores y directores. La elección de los parámetros para la creación del grafo se realizó utilizando un enfoque de prueba y error. En los resultados de la Tabla 13 se reportan los tres resultados más representativos obtenidos, mientras que la lista de parámetros utilizada se encuentra en el anexo B.

Los resultados obtenidos utilizando el grafo mejoran considerablemente al incluir los datos Pre-producción, desde un *accuracy* del 25 % al 37 %. En las diferentes versiones del grafo Pre-producción se trataron de capturar distintos relaciones, y se realizaron de forma iterativa, siguiendo el orden de las versiones reportadas. La intuición detrás de cada versión se explicará en los siguientes párrafos.

La versión 1 fue el enfoque inicial y se le dio la misma a todos los aspectos. Los resultados obtenidos son comparables a los obtenidos con los datos de Desarrollo, pero son peores que los datos genéricos, por lo tanto, no se está capturando toda la información relevante.

En la versión 2 se trató de aprovechar la información de las precuelas y películas del mismo estudio, aumentando su importancia. El motivo de aumentar la importancia se debe a que las precuelas por lo general suelen tener un comportamiento en taquilla similar a las precuelas. Por otra parte, al incrementar la importancia de las películas del mismo estudio, se produce un efecto similar a las precuelas, y permite mejorar las predicciones, por ejemplo, en las películas producidas por el estudio *Blumhouse Productions*, que a pesar de producir generalmente películas que obtienen un alto presupuesto, se clasificarían como películas de baja taquilla debido a su reducido presupuesto. Los resultados obtenidos mejoran la versión 1, pero siguen siendo inferiores a los datos genéricos.

En la versión 3, a parte de las modificaciones introducidas en la versión 2, se

aumentó la importancia de los actores y directores sin llegar a ser tan importantes como las precuelas y películas del mismo estudio, mientras que se redujo la importancia del género. La intuición de aumentar la importancia de los actores y directores es similar al argumento utilizado en la versión 2, el rendimiento de los actores y directores en películas previas es un indicativo del rendimiento en las siguientes películas. La reducción de la importancia del género se debe a que algunas de las películas que se introducían debido al género tenían un grado de similitud bajo con la película actual. Por otra parte, el género se siguió considerando porque permite dar mas importancia a las interacciones, como actores que han participado en el género de la película con anterioridad, o estudios que se centran en un tipo de género. Los resultados obtenidos mejoran la versión 2, y a pesar de no mejorar los resultados genéricos, captura relaciones que en conjunción con los datos genéricos mejoran los resultados del modelo.

Finalmente, se creo un modelo utilizando los datos genéricos, el grafo versión 3 y el modelo con variables textuales que obtuvo los mejores resultados de Desarrollo. Los resultados obtenidos mejoran considerablemente el modelo con datos genérico, por lo que tanto la representación del grafo como la información textual aportan información que no está contenida en los datos genéricos.

5. Conclusiones y líneas futuras

En el presente trabajo se ha tratado de predecir la taquilla obtenida por una película utilizando datos previos a su producción. El problema se ha tratado como clasificación, y se propusieron dos formas de realizar la discretización, la primera en 2 clases para comprobar la viabilidad del problema, y la segunda en 9 clases para facilitar la comparación con otros trabajos de la literatura. La discretización se ha realizado de tal forma que todas las clases tengan el mismo número de instancias.

El trabajo se centró en las etapas de Desarrollo, donde los productores deciden si producir una película, y Pre-producción, donde se planifica la producción de la películas, como actores, directores y localizaciones. En dichas fases los recursos invertidos son reducidos, y por lo tanto, las predicciones son útiles en la ayuda de toma de decisiones para los inversores, mientras que las predicciones con datos posteriores, no son tan relevantes debido a que ya se han invertido muchos recursos.

Tanto en la fase de Desarrollo como Pre-producción, se han realizado varios experimentos, obteniendo un *accuracy* 74.4 % y 87.2 % en la clasificación binaria, y 32.9 % y 46.9 % en la multi-clase respectivamente. La mayor cantidad de información disponible en la fase Pre-producción respecto a la fase de Desarrollo, se traduce en una considerable mejora en los resultados. El resultado más destacable, en el modelo Pre-producción multi-clase, donde se obtuvo un 46.9 % de *accuracy* y un 80.6 % de *1-away*, que es una variación de *accuracy* que considera como acierto las clases adyacentes. Este resultado es comparable con trabajos en la literatura con datos de etapas posteriores, donde la cantidad de información es superior, pero la utilidad de las predicciones es menor debido a los recursos invertidos.

Las contribuciones al estado del arte se resumen en el uso de modelos basados en *Deep Learning*, la extracción de información textual utilizando *word2vec* y Redes neuronales recurrentes (RNN), y el uso de representaciones de grafos basados en *node2vec*. El uso de estas variables ha permitido mejorar los resultados obtenidos en la fase de Desarrollo y Pre-producción un 13.7 % y 24.3 % respectivamente respecto a los experimentos realizados con las variables utilizadas normalmente en la literatura.

El uso de técnicas de Procesamiento del Lenguaje Natural basadas en *word2vec* y RNN han obtenido unos resultados ligeramente superiores a técnicas previamente utilizadas en la literatura como Frecuencia de término - frecuencia inversa de documento (TF-IDF), pero tienen un costo computacional superior.

Se ha desarrollado un método que permite la construcción de grafos, en la que los nodos representan películas y las aristas conexiones entre películas similares, en base a atributos categóricos y variables continúas. Posteriormente, se convirtieron los nodos en una representación densa utilizando *node2vec*. Esto ha permitido, con un número reducido de películas, extraer información de variables categóricas con

una gran cardinalidad, así como capturar interacciones entre las variables y aprovechar la temporalidad de las películas.

Finalmente, se ha concluido que discretizar la taquilla y tratar el problema como clasificación genera sesgos, puede ocasionar problemas de interpretación de los resultados y dificulta las comparaciones con otros trabajos de la literatura.

Como posibles líneas de trabajo futuro, se proponen las siguientes:

- Actualmente la optimización de los parámetros del grafo se está llevando a cabo de forma manual mediante un mecanismo de prueba y error, por lo que una posible linea de trabajo sería automatizar dicha optimización. Una posible opción sería optimizarlo mediante el uso de algoritmos genéticos, donde cada individuo representaría un conjunto de parámetros para la creación del grafo. Para el cálculo del *fitness* del individuo, se propone calcular una estimación del presupuesto de la película realizando una media ponderada, utilizando los pesos de las aristas, de los presupuestos de las películas a las que está conectada una determinada película. El *fitness* del individuo se calcularía como el error absoluto medio de las películas marcadas como entrenamiento, y el objetivo sería minimizar dicho valor.

- En la construcción del grafo sólo se está utilizando información de la fase de Desarrollo y Pre-producción, pero debido a la división temporal de los datos de entrenamiento y prueba, y que los datos de prueba únicamente se conectan con los de entrenamiento, se podrían utilizar datos Post-producción en las observaciones de entrenamiento. También se podrían introducir conexiones a películas similares en base al criterio de un experto, al igual que se realiza en las valoraciones manuales en la fase de Desarrollo.

- Además, la construcción del grafo se podría realizar de forma iterativa, introduciendo la predicción de la taquilla del modelo como atributo. De esta forma, se podría utilizar la taquilla para minimizar el número de conexiones con película con taquillas dispares y mejorar las predicciones.

- Respecto a la información textual, la principal limitación de los modelos basados en RNN es el reducido número de películas. Una posible solución sería buscar un dominio similar, y utilizar *transfer-learning*, que es una técnica en la que un modelo se entrena para una tarea pero después se aplica a un dominio similar. Esta técnica es ampliamente utilizada en otros dominios como el reconocimiento de objetos, y permitiría utilizar modelos más grandes, y por lo tanto, capturar relaciones más complejas.

Referencias

[1] Attention mechanism. URL `https://gist.github.com/cbaziotis/`. Accessed: 2018-05-27.

[2] Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[3] Khalid Ibnal Asad, Tanvir Ahmed, and Md. Saiedur Rahman. Movie popularity classification based on inherent movie attributes using c4.5,part and correlation coefficient. *CoRR*, abs/1209.6070, 2012. URL `http://arxiv.org/abs/1209.6070`.

[4] S. Asur and B. A. Huberman. Predicting the future with social media. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 492–499, 2010. doi: 10.1109/WI-IAT.2010.63.

[5] David Bamman, Brendan O'Connor, and Noah A Smith. Learning latent personas of film characters. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, page 352, 2014.

[6] Rounak Banik. The movies dataset. https://www.kaggle.com/rounakbanik/the-movies-dataset.

[7] Suman Basuroy and Subimal Chatterjee. Fast and frequent: Investigating box office revenues of motion picture sequels. *Journal of Business Research*, 61(7):798 – 803, 2008. ISSN 0148-2963. doi: https://doi.org/10.1016/j.jbusres.2007.07.030. URL `http://www.sciencedirect.com/science/article/pii/S0148296307002482`.

[8] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[9] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3 (Feb):1137–1155, 2003.

[10] Steven Bird and Edward Loper. Nltk: the natural language toolkit. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 31. Association for Computational Linguistics, 2004.

[11] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

[12] Yuanyuan Chen, Yisheng Lv, Xiao Wang, and Fei-Yue Wang. A convolutional neural network for traffic information sensing from social media text. In *Intelligent Transportation Systems (ITSC), 2017 IEEE 20th International Conference on*, pages 1–6. IEEE, 2017.

[13] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[14] François Chollet et al. Keras, 2015.

[15] Dan Cocuzzo and Stephen Wu. Hit or flop: Box office prediction for feature films. *Stanford University*, 2013.

[16] Dursun Delen and Ramesh Sharda. Predicting the financial success of hollywood movies using an information fusion approach. *Indus Eng J*, 21(1):30–37, 2010.

[17] Guanghuiand Weinberg Charles B. Dhar, Tirthaand Sun. The long-term box office performance of sequel movies. *Marketing Letters*, 23(1):13–29, Mar 2012. ISSN 1573-059X. doi: 10.1007/s11002-011-9146-1. URL `https://doi.org/10.1007/s11002-011-9146-1`.

[18] Jingfei Du, Hua Xu, and Xiaoqiu Huang. Box office prediction based on microblog. *Expert Systems with Applications*, 41(4, Part 2):1680 – 1689, 2014. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2013.08.065. URL `http://www.sciencedirect.com/science/article/pii/S0957417413006866`.

[19] Jehoshua Eliashberg, Sam K Hui, and Z John Zhang. From story line to box office: A new approach for green-lighting movie scripts. *Management Science*, 53(6):881–893, 2007.

[20] Jehoshua Eliashberg, Sam K Hui, and Z John Zhang. Assessing box office performance using movie scripts: A kernel-based approach. *IEEE Transactions on Knowledge and Data Engineering*, 26(11):2639–2648, 2014.

[21] Stephen Follows. Follows, stephen. "does hollywood use the same movie release pattern every year?" the spread, 30 mar. 2016, cinemajam.com/mag/features/hollywood-release-patterns., Apr 2016. URL `http://cinemajam.com/mag/features/hollywood-release-patterns`.

[22] M. Ghiassi, David Lio, and Brian Moon. Pre-production forecasting of movie revenues with a dynamic artificial neural network. *Expert Systems with Applications*, 42(6):3176 – 3193, 2015. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2014.11.022. URL `http://www.sciencedirect.com/science/article/pii/S0957417414007088`.

[23] Goetzmann, William N., Ravid, and Ronald S. Abrahamand Sverdlove. The pricing of soft and hard information: economic lessons from screenplay sales. *Journal of Cultural Economics*, 37(2):271–307, 2013. doi: 10.1007/s10824-012-9183-5. URL https://doi.org/10.1007/s10824-012-9183-5.

[24] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016.

[25] Richard HR Hahnloser and H Sebastian Seung. Permitted and forbidden sets in symmetric threshold-linear networks. In *Advances in Neural Information Processing Systems*, pages 217–223, 2001.

[26] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[28] Mark B.and Walsh Gianfranco Hennig-Thurau, Thorstenand Houston. Determinants of motion picture box office and profitability: an interrelationship approach. *Review of Managerial Science*, 1(1):65–92, Apr 2007. ISSN 1863-6691. doi: 10.1007/s11846-007-0003-9. URL https://doi.org/10.1007/s11846-007-0003-9.

[29] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[30] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[31] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251 – 257, 1991. doi: https://doi.org/10.1016/0893-6080(91)90009-T. URL http://www.sciencedirect.com/science/article/pii/089360809190009T.

[32] Starling Hunter. A novel method of network text analysis. 04:350–366, 01 2014.

[33] Starling Hunter, Susan Smith, and Saba Singh. Predicting box office from the screenplay: A text analytical approach. 7:135–154, 06 2016.

[34] Minhoe Hur, Pilsung Kang, and Sungzoon Cho. Box-office forecasting based on sentiments of movie reviews and independent subspace method. *Information Sciences*, 372:608 – 624, 2016. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2016.08.027. URL http://www.sciencedirect.com/science/article/pii/S0020025516306016.

[35] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[36] Mahesh Joshi, Dipanjan Das, Kevin Gimpel, and Noah A Smith. Movie reviews and revenues: An experiment in text regression. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 293–296. Association for Computational Linguistics, 2010.

[37] Michael T. Lash and Kang Zhao. Early predictions of movie success: The who, what, and when of profitability. *Journal of Management Information Systems*, 33(3):874–903, 2016. doi: 10.1080/07421222.2016.1243969. URL https://doi.org/10.1080/07421222.2016.1243969.

[38] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521 (7553):436, 2015.

[39] Kyuhan Lee, Jinsoo Park, Iljoo Kim, and Youngseok Choi. Predicting movie success with machine learning techniques: ways to improve accuracy. *Information Systems Frontiers*, pages 1–12, 2016.

[40] Barry R. Litman. Predicting success of theatrical movies: An empirical study. *The Journal of Popular Culture*, 16(4): 159–175, 1983. doi: 10.1111/j.0022-3840.1983.1604_159.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.0022-3840.1983.1604$_1$59.x.

[41] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013.

[42] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[43] Wes McKinney. Data structures for statistical computing in python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 51 – 56, 2010.

[44] Márton Mestyán, Taha Yasseri, and János Kertész. Early prediction of movie box office success based on wikipedia activity big data. *PloS one*, 8(8):e71226, 2013.

[45] Marvin Minsky and Seymour A Papert. *Perceptrons: an introduction to computational geometry*. 1969.

[46] MPAA. Theatrical market statistics 2016. 2016. URL https://www.mpaa.org/wp-content/uploads/2018/03/MPAA-Theatrical-Market-Statistics-20
1.*pdf*.

[47] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[48] Robert Nelson, Randy A.and Glotfelty. Movie stars and box office revenues: an empirical analysis. *Journal of Cultural Economics*, 36(2):141–166, May 2012. ISSN 1573-6997. doi: 10.1007/s10824-012-9159-5. URL `https://doi.org/10.1007/s10824-012-9159-5`.

[49] Travis E Oliphant. *A guide to NumPy*, volume 1. Trelgol Publishing USA, 2006.

[50] Doina Parimi, Rohitand Caragea. Pre-release box-office success prediction for motion pictures. In Petra Perner, editor, *Machine Learning and Data Mining in Pattern Recognition*, pages 571–585, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-39712-7.

[51] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[52] James Prag, Jayand Casavant. An empirical study of the determinants of revenues and marketing expenditures in the motion picture industry. *Journal of Cultural Economics*, 18(3):217–235, Sep 1994. ISSN 1573-6997. doi: 10.1007/BF01080227. URL `https://doi.org/10.1007/BF01080227`.

[53] N. Quader, M. O. Gani, D. Chaki, and M. H. Ali. A machine learning approach to predict movie box-office success. In *2017 20th International Conference of Computer and Information Technology (ICCIT)*, pages 1–7, Dec 2017.

[54] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

[55] T. G. Rhee and F. Zulkernine. Predicting movie box office profitability: A neural network approach. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 665–670, Dec 2016. doi: 10.1109/ICMLA.2016.0117.

[56] M Saraee, S White, J Eccleston, et al. A data mining approach to analysis and prediction of movie ratings. *Transactions of the Wessex Institute*, pages 343–352, 2004.

[57] Ramesh Sharda and Dursun Delen. Predicting box-office success of motion pictures with neural networks. *Expert Systems with Applications*, 30(2):243 – 254,

2006. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2005.07.018. URL http://www.sciencedirect.com/science/article/pii/S0957417405001399.

[58] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[59] Josef Steiff. *The complete idiot's guide to independent filmmaking*. Penguin, 2005.

[60] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–656, 2015.

[61] Matt Vitelli. Predicting box office revenue for movies. 2015.

[62] Max Wasserman, Satyam Mukherjee, Konner Scott, Xiao Han T. Zeng, Filippo Radicchi, and Luís A. N. Amaral. Correlations between user voting data, budget, and box office for films in the internet movie database. *Journal of the Association for Information Science and Technology*, 66(4):858–868, 2014. doi: 10.1002/asi.23213. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.23213.

[63] Li Zhang, Jianhua Luo, and Suying Yang. Forecasting box office revenue of movies with bp neural network. *Expert Systems with Applications*, 36(3, Part 2):6580 – 6587, 2009. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2008.07.064. URL http://www.sciencedirect.com/science/article/pii/S095741740800496X.

[64] Z. Zhang, J. Chai, B. Li, Y. Wang, M. An, and Z. Deng. Movie box office inteval forecasting based on cart. In *2015 8th International Symposium on Computational Intelligence and Design (ISCID)*, volume 2, pages 87–90, Dec 2015. doi: 10.1109/ISCID.2015.165.

[65] Yao Zhou, Lei Zhang, and Zhang Yi. Predicting movie box-office revenues using deep neural networks. *Neural Computing and Applications*, pages 1–11, 2017.

A. Librerías utilizadas

El entorno de programación utilizado fue python debido a que es uno de los entornos más populares para aprendizaje automático y cuenta con una gran comunidad, lo que se traduce en un gran número de librerías y recursos disponibles. Otra de las ventajas de python, es la simplicidad de su código, que facilita la reproducibilidad y reutilización de código por parte de otros investigadores.

Las librerías utilizadas en el desarrollo del trabajo son las siguientes:

- Tensorflow [2]: Tensorflow es una librería para computación numérica representada mediante grafos aciclicos dirigidos haciendo hincapié en el desarrollo de redes neuronales profundas.
- nltk [10]: Es una colección de herramientas para tareas de lenguaje natural.
- Keras [14]: Keras es una capa de abstracción orientada a la facilidad de uso sobre algunos de los frameworks de redes neuronales profundas como Tensorflow.
- pandas [43]: Es una librería para manipulación y análisis de datos en matrices y listas.
- numpy [49]: Es una librería para el manejo de listas multi-dimensionales y matrices.
- sklearn [51]: Es un librería de aprendizaje automático para python.
- gensim [54]: Es una librería para el procesamiento del lenguaje natural.

B. Parámetros utilizados en la creación del Grafo

Aspecto	max_con	aspect_weight
Escritores	20	0.25
género	100	0.05
Estudio cinematográfico	20	0.2
Precuelas	5	0.5

Tab. 14: Parámetros utilizados para la creación del grafo de Desarrollo.

Aspecto	max_con	aspect_weight
Escritores	20	0.16
género	70	0.16
Estudio cinematográfico	20	0.16
Precuelas	5	0.16
Actores	30	0.16
Directores	10	0.16

Tab. 15: Parámetros utilizados para la creación del grafo de Pre-producción v1.

Aspecto	max_con	aspect_weight
Escritores	20	0.11
género	100	0.11
Estudio cinematográfico	20	0.30
Precuelas	5	0.25
Actores	30	0.11
Directores	10	0.11

Tab. 16: Parámetros utilizados para la creación del grafo de Pre-producción v2.

Aspecto	max_con	aspect_weight
Escritores	20	0.11
género	100	0.05
Estudio cinematográfico	20	0.29
Precuelas	5	0.22
Actores	30	0.18
Directores	10	0.15

Tab. 17: Parámetros utilizados para la creación del grafo de Pre-producción v3.