INFORME PROYECTO

Deep Learning
Movie Analyser

I. Contexto

El éxito comercial y crítico de una película es un fenómeno complejo influenciado por múltiples factores: el argumento, el género, el director, el reparto, el presupuesto y las tendencias culturales de cada época. Sin embargo, ciertos patrones narrativos y estilísticos pueden correlacionarse con un mayor atractivo para el público. En este proyecto, nos enfocamos en analizar sinopsis, géneros y directores como variables principales para entrenar un modelo de deep learning capaz de predecir el potencial de éxito de una película. Este enfoque no solo puede ser útil para estudios de cine y productores al evaluar proyectos en fase de desarrollo, sino también para plataformas de streaming que deseen anticipar la recepción de sus contenidos.

II. Estructura de los notebooks

En esta parte, nos vamos a explicar la estructura de los notebooks, como se utilizan y como han evaluado.

1. 01_exploracion_de_los_datos

Este primer notebook está enfocado a descubrir los datos que vamos a utilizar y los potenciales problemas que vamos a encontrar. Esta compartido en dos partes, una parte para el dataset IMDB Movie y otra parte para el dataset FilmTV.

Primero, usamos dos datasets por múltiples razones. La primera razón es que son dos dataset de tamaño muy diferente (un pequeño, 1000 valores, y un mucho más grande, 40000 valores). Así, cuando queremos probar una nueva arquitectura de modelo, la podemos probar con el pequeño dataset incluso si no es perfecto. La segunda razón es porque eso nos permite de probar un modelo entrenado por un dataset con el otro dataset y ver sus resultados que así no pueden ser sesgados.

En este notebook entonces, nos enfocamos primero con la descargada del primer dataset (IMDB Movies). Luego, mezclamos los datos porque hemos visto que fueran ordenadas por su valor de éxito y eso habría sido un problema si habríamos entrenado nuestro modelo así.

Después, vemos que algunos datos tienen un valor NaN según el atributo. Así que nos acordaremos de borrar esas líneas que tienen al menos un valor NaN.

Finalmente, nos damos cuenta de que los valores de las notas son muy agrupados y que eso será un problema para nuestro modelo porque le faltará diversidad para que se entrena correctamente. Esas razones reforzaron mi creencia en encontrar un segundo dataset.

Por la segunda parte de este notebook, exploramos el segundo dataset, FilmTV. Como lo dijimos, este dataset es 40 veces mas grande que el primero. En primer lugar, le descargamos y después, mostramos los datos que vamos a usar para nuestro modelo (hay las mismas informaciones que nos interesa y tienen el mismo formato que antes).

Enseguida, vemos que la amplitud de las notas de las películas es mucho mas grande incluso si la mayoría de esas notas esta centrada.

Finalmente, vemos que, como para el primer dataset, este dataset tiene a veces valores NaN que deberemos borrar.

2. 02_preprocesado

Este segundo notebook está enfocado en transformar los datos brutos en un formato adecuado para entrenar un modelo de deep learning. Mientras que el primer notebook se centraba en explorar y comprender las características de cada dataset, este notebook define de manera clara el pipeline de preparación de los datos, asegurando consistencia entre IMDB y FilmTV y eliminando cualquier ruido que pudiera afectar el rendimiento del modelo.

En primer lugar, eliminamos todas las filas que contienen al menos un valor NaN. Esta decisión se basa en el hecho de que los atributos eliminados (director, género, sinopsis o nota) son esenciales para el modelo, y no sería coherente sustituirlos o imputarlos de manera artificial.

A continuación, aplicamos un preprocesado específico para cada uno de los atributos que utilizará el modelo:

- **Overview (sinopsis)**
  Aplicamos una limpieza básica eliminando los stopwords en ingles únicamente. Luego tokenizamos la sinopsis con un Tokenizer limitado a un número ajustado de términos. Cada sinopsis se transforma en una secuencia numérica y finalmente se aplica padding para asegurar una longitud uniforme.

- **Título**
  El título también es tokenizado como texto, ya que algunos títulos están formados por varias palabras. En el caso de FilmTV, los Titulos pueden estar escritas en diferentes idiomas, por lo que construimos un conjunto de stopwords multilingüe (inglés, español, francés, italiano y alemán) y les borramos de los títulos. Finalmente, utilizamos un Tokenizer dedicado y aplicamos padding para obtener una representación consistente entre películas.

- **Director**
  Para que un mismo director sea reconocido consistentemente a lo largo de todas sus películas, utilizamos un LabelEncoder. De esta manera, cada director recibe un identificador único, que luego será convertido en embedding en el modelo. Así, el modelo puede aprender una representación densa y significativa asociada a cada director.

- **Género**
  El género puede contener uno o varios valores (por ejemplo, “Drama”, “Action, Thriller”). Por ello, separamos los géneros, convertimos cada conjunto de géneros en una

secuencia de tokens y aplicamos padding. Esta elección permite al modelo tener una representación multietiqueta flexible.

- **Nota o puntuación**
  En el caso del dataset IMDB, combinamos la nota IMDB y el metascore en una sola puntuación mediante un promedio ponderado. Para ambos datasets, la nota final se normaliza entre 0 y 1, lo cual facilita la tarea de regresión del modelo.

Finalmente, dividimos los datos en un conjunto de entrenamiento (80%) y un conjunto de prueba (20%), asegurando que cada entrada está compuesta por:

- overview tokenizado
- título tokenizado
- géneros tokenizados
- identificador de director
- puntuación normalizada

Este notebook genera todos los tensores finales que utiliza nuestro modelo de deep learning.

3. 03_arquitectura_de_linea_de_base

Este ultimo notebook contiene finalmente nuestras soluciones, los modelos. Primero, con este notebook, podemos desplegar el dataset que queremos (IMDB o FilmTV). Luego, hay el procesamiento de los datos que ya hemos visto.

Finalmente llegan los modelos. Hay 5 modelos para cinco iteraciones que veremos en otra parte de este informe.

Cada iteración se puede ejecutar independiente de las otras y todas tienen la misma forma. Primero, creamos el modelo con sus modificaciones. Luego, le hacemos un build y vemos su resumen con sus pesos. Después, entretenemos el modelo con los datos que hemos elegido. Por fin, analizamos los resultados del modelo con diferentes métricas como: el MAE, el RMSE o el R^2. También, comparamos unas predicciones con el valor real de la película. Por fin, visualizamos en una tabla con los valores reales y previstos la eficacidad de nuestro modelo.

III. Solución

1. Arquitectura

La arquitectura final del proyecto se basa en un modelo híbrido compuesto por varias ramas y centrado en un **Transformer** para procesar la sinopsis, que es la fuente de información más rica. Para el *overview*, utilizamos *Positional Embedding* seguido de un **bloque Transformer Encoder** con multi-head attention y pooling global, lo que permite capturar dependencias largas y patrones semánticos complejos.

El *título* se procesa mediante un **mini-Transformer**, más ligero y adaptado a secuencias cortas. Los *géneros* y el *director* se representan mediante *embeddings* y pequeños bloques densos, suficientes para este tipo de información categórica.

Las cuatro ramas se fusionan y pasan por un bloque denso con *Batch Normalization* y *Dropout*, que actúa como regresor final.

Esta arquitectura ofrece mejor estabilidad y mayor capacidad de representación que los modelos recurrentes utilizados en las iteraciones anteriores.

2. Preprocesado

El preprocesado tiene como objetivo transformar los datos brutos de ambos datasets (IMDB y FilmTV) en tensores compatibles y coherentes para alimentar el modelo de deep learning. Primero eliminamos todas las filas que contienen valores *NaN*, ya que los atributos incompletos (director, género, sinopsis o nota) son esenciales para el aprendizaje.

A continuación, aplicamos un tratamiento específico a cada atributo:

- **Overview (sinopsis)**:
  Limpieza básica eliminando stopwords en inglés, seguida de tokenización y padding para obtener secuencias de longitud uniforme.
- **Título**:
  Se eliminan stopwords multilingües (inglés, español, francés, italiano, alemán), luego se tokeniza y se aplica padding para asegurar una representación homogénea.
- **Director**:
  Cada director recibe un identificador único mediante *LabelEncoder*, que luego será convertido en embeddings dentro del modelo.
- **Género**:
  Los géneros múltiples se separan, se tokenizan como secuencias y se aplica padding, permitiendo una representación multietiqueta flexible.
- **Puntuación**:
  En IMDB combinamos la nota IMDB y el metascore mediante un promedio ponderado. En ambos datasets, la puntuación final se normaliza entre 0 y 1 para facilitar la regresión.

Finalmente, dividimos los datos en entrenamiento (80%) y prueba (20%), generando los tensores finales que servirán como entrada del modelo.

IV. Iteraciones

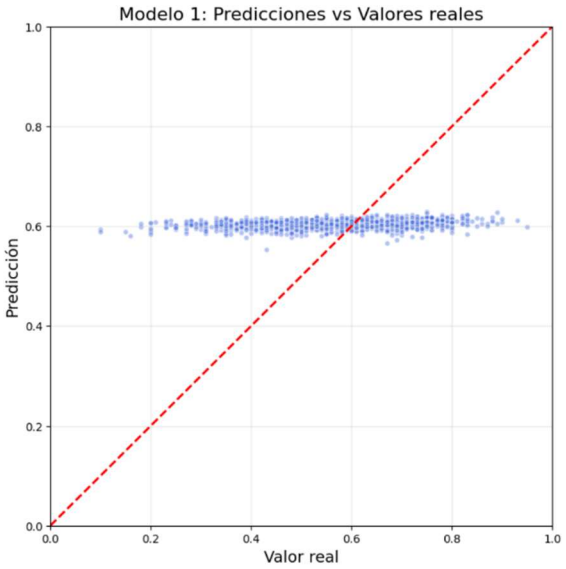
Durante el desarrollo del proyecto se construyeron **cinco iteraciones de modelos**, cada una incorporando mejoras progresivas tanto en la arquitectura como en el tratamiento de los datos. Todas las iteraciones comparten la misma estructura general de evaluación, pero difieren en complejidad y capacidad de representación.

1. Iteración 1 — Modelo Base Simple

La primera versión sirvió como línea de base.

- El overview se procesaba mediante un **embedding + GlobalAveragePooling1D**, sin mecanismos de atención.

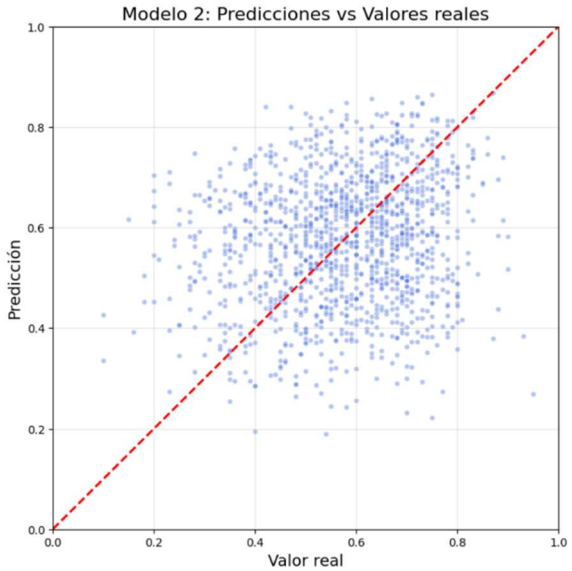
- El rendimiento fue limitado debido a la incapacidad del modelo para capturar dependencias semánticas de la sinopsis.



2. Iteración 2 — LSTM

En esta iteración se añadió:

- Una capa RNN LSTM para capturar mas informaciones semanticas.
- Mejoras que redujeron el overfitting, pero seguían sin capturar relaciones complejas en el texto.

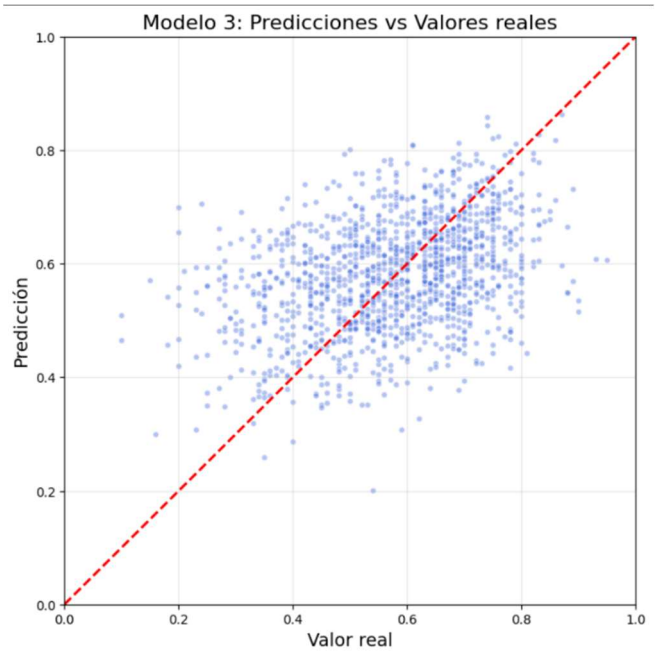


3. Iteración 3 — Primer uso de mecanismos de atención

Aquí se introdujo por primera vez:

- Los otros parámetros como el título, el genero o el director.

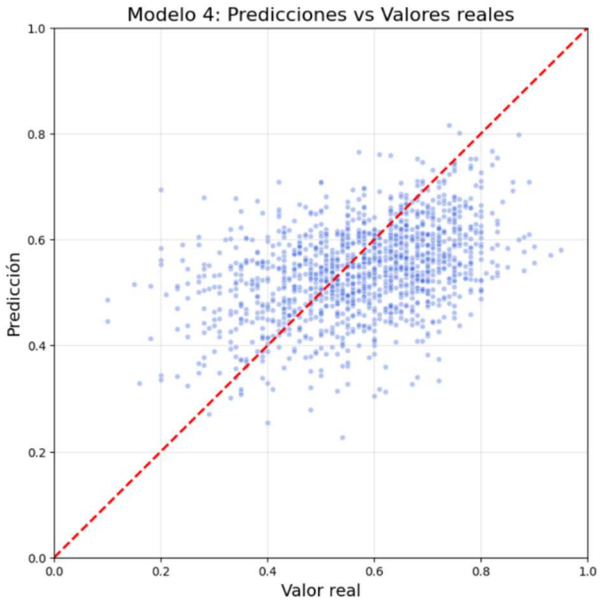
- Un procesamiento más rico en general, detectando más informaciones relevantes. Este modelo ya mostró una mejora notable respecto a los dos anteriores.



4. Iteración 4 — Transformer Encoder para la sinopsis

Este modelo 4 introduzco muchas cosas:

- Capas de Dropouts y de BatchNormalization
- Cambio las funciones de activación para gelu que es mejor
- Cambio las capas LSTM para GRU que nos permite de entrenar nuestro modelo con más epochs
- Un callback EarlyStopping para evitar el overfitting



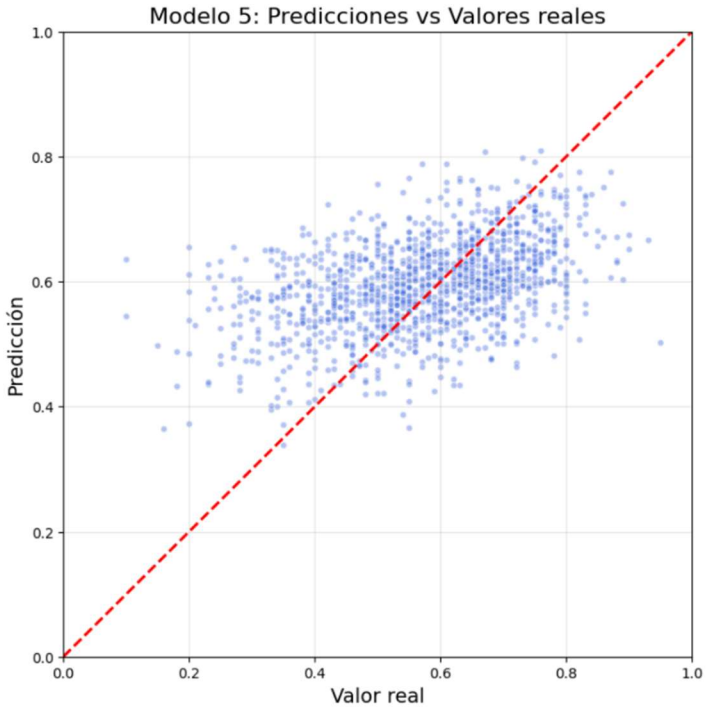
5. Iteración 5 — Arquitectura Final (Modelo Híbrido Completo)

Esta última versión marca un salto importante:

- Se implementó un **Transformer Encoder completo** en el overview, con multi-head attention.
- Se mejoró la rama del título con pequeño Transformer.
- Se uso para esas dos ultimas capas un **PositionnalEmbedding** por la primera vez.

Este modelo consiguió capturar dependencias largas y semántica contextual.

Pero, esta versión no marca un salto tan importante:



V. Resultados

Se evaluaron los modelos utilizando tres métricas principales:

- **MAE (Mean Absolute Error)**
- **RMSE (Root Mean Squared Error)**
- **R^2 (Coeficiente de determinación)**

Los resultados muestran una clara progresión positiva conforme aumenta la complejidad del modelo.

Análisis cualitativo

- El **modelo base** no distingue correctamente películas de éxito bajo o alto.
- La introducción de **self-attention** mejora la sensibilidad respecto a sinopsis largas.

- Los modelos con capas RNN y todos los inputs (iteración 3, 4 y 5) nos permitieron de mejorar significamente los resultados, pero después de eso, tomara mucho mas tiempo para mejorar mucho el modelo.

Comprobación manual

En todas las iteraciones se compararon predicciones puntuales con valores reales mediante tablas generadas en el notebook.

VI. Resumen

El proyecto Movie Analyser logró construir un sistema de deep learning capaz de predecir el éxito de una película partiendo de su sinopsis, género, título y director.

El flujo completo del trabajo fue:

1. **Exploración rigurosa** de dos datasets complementarios (IMDB y FilmTV).
2. **Preprocesamiento estructurado** y coherente entre ambos, con técnicas modernas de preparación de texto.
3. **Diseño iterativo de modelos**, desde arquitecturas simples hasta un Transformer completo.
4. **Un modelo final híbrido**, con:
    - Transformer para sinopsis,
    - mini-Transformer para título,
    - embeddings optimizados para director y género,
    - regresión final regularizada.

Este proyecto me permitió de ejercer lo que aprendí este semestre, pero, por una mala gestión de mi tiempo y un fin de semestre muy cargado, no pude llevar este proyecto a resultados satisfechos.